TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Jirui Yuan

Robust Face Recognition under Uncontrolled Environments

Julkaisu 1561 • Publication 1561

Tampere 2018

Tampereen teknillinen yliopisto. Julkaisu 1561
Tampere University of Technology. Publication 1561

Jirui Yuan

Robust Face Recognition under Uncontrolled Environments

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB 109, at Tampere University of Technology, on the 31st of August 2018, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2018

Doctoral candidate: Jirui Yuan
Laboratory of Signal Processing
Faculty of Computing and Electrical Engineering
Tampere University of Technology
Finland

Supervisor: Karen Egiazarian (Eguiazarian), Professor
Laboratory of Signal Processing
Faculty of Computing and Electrical Engineering
Tampere University of Technology
Finland

Pre-examiners: Edward J. Delp, Professor
School of Electrical and Computer Engineering
Purdue University
United States

Nicola Conci, Associate Professor
Department of Information Engineering and Computer Science
University of Trento
Italy

Opponent: Guoying Zhao, Professor
Center for Machine Vision and Signal Analysis
University of Oulu
Finland

ISBN 978-952-15-4179-7 (printed)
ISBN 978-952-15-4185-8 (PDF)
ISSN 1459-2045

Abstract

The performance of face recognition suffers from content-noise and label noise in datasets, and from insufficient and unlabeled datasets. This thesis starts from exploring robust face representation descriptors and proposes shallow face alignment and face recognition methods that tackles the problem of content-noise in datasets. The alignment method is proposed and proven to be superior to the prior art in terms of better accuracy in case of noise presence with a neglectable computational complexity increase and with the possibility of its parallel implementation on GPU. In the case of unlabeled datasets, in this thesis, a multi-view predictive (MVP) latent space learning model is proposed for multi-view unsupervised learning. Experimental results show that MVP improves the performance of multi-view clustering. To further explore face representation models that are robust to content-variations in dataset, co-regularized sparse representation of Gaussians based classifier is proposed. It is demonstrated that the proposed method outperforms the state-of-the-art algorithms.

Deep models have achieved a consistent breakthrough in face recognition**.** The superior performance of deep learning owns to the representations of data with multiple levels of abstraction and massive labelled training data. To acquire a clean face dataset, a robust deep face recognition method is proposed using automatic outlier removal. Other than a label noise in a massive dataset, there are also challenges when the dataset has insufficient samples. This thesis explores the possibility of transferring tasks on other large-scale dataset to current task on a limited-scale dataset.

Extensive experiments are carried out and show that the methods proposed in this thesis alleviate above-mentioned noise problems that are existed in the uncontrolled real-world environments.

Preface

The research work on which this thesis is based has been performed during my stay at the Department of Signal Processing, Tampere University of Technology (TUT), Finland.

This work has been accomplished under the supervision of Prof. Karen Egiazarian. I am greatly indebted towards him, for his highly professional guidance and for being so patient with me. Without his precious support, my research work would have not progressed to this point.

I sincerely thank the pre-examiners of my thesis, Prof. Edward J. Delp and Dr. Nicola Conci for the careful assessment of my work and for the valuable comments.

I wish to express my gratitude to Prof. Hexin Chen and Prof. Yan Zhao for recommending me the chance to pursue my research studies in Finland, and to Prof. Shiguang Shan for supporting me when I remotely work on my PhD thesis in China.

Finally, I would like to express my special thanks to my mother Suping Nian for her love and endless supports throughout my life. My special thanks to my husband Hongwei Meng for supporting and understanding me during my study. Most of all, deepest thanks to our dearest daughter Pinyan for all the joys she brought to me and for letting me be a mother.

Tampere, August 2018

Jirui Yuan

Contents

List of Figures

List of Symbols and Abbreviations

$\boldsymbol{x}$	image data vector
$I(\boldsymbol{x})$	2D grey-scale image
$T(\boldsymbol{x})$	2D grey-scale template
$\eta(\boldsymbol{x})$	noise
h	scale
θ_i	direction
g_{h,θ_i}	kernel for direction θ_i and scale h
w_{h,θ_i}	directional window
ϕ_h	complete set of linearly independent 2D polynomials
m	the order of polynomial
M	the length of the vector
$\circledast$	convolution operator
Q_j	confidence interval
L_j	lower bound
U_j	upper bound
Γ	threshold
$\hat{I}$	directional estimate of I
$\hat{T}$	directional estimate of T
σ	standard variation
$\boldsymbol{p}$	parameters vector
$\Delta\boldsymbol{p}$	increment to the parameters $\boldsymbol{p}$
$\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$	the parameterized set of allowed warps
$\nabla\hat{I}_\theta$	the gradient of the image $\hat{I}_\theta$
L	a generic pyramidal level
I^l	image at level l
n_x^l	width of I^l
n_y^l	height of I^l
D	direction set
h_{ol}	half height of overlapping areas
w_{ol}	half width of overlapping areas
g_c	central pixel
$mLBP_M(g_c)$	MLBP of g_c
$s(x)$	sign function
$D_{h,\theta_i}^l(x)$	directional derivative for direction θ_i and scale h
$M_{\theta_i}^l$	mLBP map
r^l	partitioning level

R^l	maximum partitioning level of r^l
$h_{i,r}^l$	histograms
w_r^l	weight vector
$H_{\theta_i}^l$	descriptor of l[th] level in direction θ_i
H^l	descriptor of l[th] level
H	descriptor
$\{\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_n\}$	image set data
$\mathbf{G}(\mathbf{u}, \mathbf{C})$	Gaussian descriptor
$\mathbf{u}$	mean value
$\mathbf{C}$	covariance matrix
w	parameter weight
$\varphi(\cdot)$	map u of Gaussians
$\phi(\cdot)$	map C of Gaussians
$\mathbf{U}$	the dictionary of mean vector
$\mathbf{V}$	the dictionary of covariance matrices
$vec(\cdot)$	vectorization operation
$\log(\cdot)$	logarithm operation
$\mathbf{a_u}$	representation coefficients of mean vectors
$\mathbf{a_c}$	representation coefficients of covariance matrices
e_i	the reconstruction error of the i[th] class
$\mathbb{R}$	1-dimension space
$\mathbb{R}^d$	d-dimension space
κ	kernel function
α_i	expansion coefficient
$\mathcal{Y}$	soft label assignment
c^+	positive value
c^-	negative value
y	the vector representation of $\mathcal{Y}$
γ	trade-off parameter
L_A	loss of task A
C_A	series of task A
N_A	number of images in one mini-batch of task A
d_i	the feature dimension of the i^{th} view
$\mathbf{U}$	latent space
$f(\mathbf{X}_i, \mathbf{U})$	the correlation between $\mathbf{X}_i$ and $\mathbf{U}$

$\boldsymbol{v}$	linear projection vector
$\boldsymbol{u}$	each column $\boldsymbol{u}$ of **U**
$\boldsymbol{\Lambda}_i$	refers to $X_i\left(X_i{}^T X_i\right)^{-1} X_i{}^T$
$tr(\cdot)$	trace of matrix
$\boldsymbol{L}_i^*$	the Laplacian matrix for the i^{th} view
w_i	the weight for the i^{th} view
α	a tradeoff parameter
Δ_i^*	normalized $\boldsymbol{\Lambda}_i$
λ	Lagrange multiplier
ξ	Lagrange multiplier
r	exponential parameter
$\boldsymbol{P}_i$	projection matrix
$\widehat{\boldsymbol{P}_i}$	a closed-form solution of $\boldsymbol{P}_i$

List of Publications

Most of the material presented in this thesis appears in the following publications (sorted according to the thesis presentation order):

I. Yuan J, Egiazarian K. Anisotropic multi-scale Lucas-Kanade pyramid. Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V. 2011, 7881(1):78810O-78810O-12.

II. Mehta R, Yuan J, Egiazarian K. Face recognition using scale-adaptive directional and textural features. Pattern Recognition, 2014, 47(5): 1846-1858.

III. Jirui Yuan, Ke Gao, Pengfei Zhu, Karen Egiazarian. Multi-view Predictive Latent Space Learning. Pattern Recognition Letters, 2018, https://doi.org/10.1016/j.patrec.2018.06.022.

IV. Jirui Yuan, Hao Cheng, Karen Egiazarian. Co-regularized Sparse Representation of Gaussians for Pattern Classification. In submission to Pattern Recognition.

V. Yuan J, Ma W, Zhu P, Egiazarian K. Robust Deep Face Recognition with Label Noise. International Conference on Neural Information Processing. Springer, Cham, 2017: 593-602.

VI. Yuan J, Ma W, Zhu P, Egiazarian K. Multi-task Deep Face Recognition. Chinese Conference on Biometric Recognition. Springer, Cham, 2017: 183-190.

Not everything from these publications is included here. Some topics were purposely discarded as they were only marginally relevant to the general idea of the presented approach. Instead, here we expand those aspects that are believed to be more useful for a clear understanding of the proposed method, and, with the same intention, we add new material and considerations that have not been published.

1. Introduction

1.1 Motivation and Scope

One fundamental task in artificial intelligence and computer vision is to understand and mimic the powerful ability of human visual system to recognize the world. As a specific research topic and sub-problem in computer vision, the task of face recognition is to analyze and judge the identity of the input face image, and so, to mimic the powerful ability of a human to automatically discriminate different people.

In a classical face recognition system, we have some face images with given identity labels and build our model based on these given data. After training, the model can be used to identify the new face images by searching for the most similar image in the training data or by matching with another test image. The process of automatic face recognition contains several steps from the input of face image to the output of identity label. Specifically, there are following steps in this process: face detection, face alignment, face representation and face feature classification.

- Face detection is performed first to decide the position of face in the image and the corresponding face image is always cropped to feed the recognition system.
- With the cropped face image, face alignment is always performed by an affine transformation to weaken the effect of face deformation induced by poses, facial expressions, partial occlusions and so on.
- Face representation is a crucial step to extract the most distinguishing features to represent the face image, which directly affects the performance of recognition. In some methods, dimension reduction is followed to make the features more compact and discriminant. In recent years, deep neural network based methods are used to automatically learn the distinguished features and have been proven to be successful for face representation.
- The last step is to learn the classification model based on the feature representation of the input face images. Before the widespread use of deep learning methods, one of the mostly used method was Support Vector Machine (SVM) [134]. It considers a face recognition as a multi-label problem to obtain multiple optimal hyper-planes to perform a classification. With the development of large scale graphics processing units (GPU) and the appearance of several

large-scale face datasets, deep neural network based methods have been increasingly applied for face recognition and also made much progress for face recognition.

Among all these steps, face representation is the most crucial one. There are mainly three types of face representation methods: global feature based, local feature based and learning based face recognitions. In the following sub-sections, face recognition methods using these three types of representations are reviewed.

1.1.1 Face Recognition with Global Feature based Face Representation

Before deep methods emerge, many different approaches have been developed for face representation. A class of methods, called holistic, considers a face image as a whole and extracts the global features from it. The most popular holistic techniques are Eigenface [59] and Fisherface [60], which are based on Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), respectively. The first method projects the image into a subspace where the individual components are ranked according to their variances. The face image is reconstructed by a linear combination of these components that satisfy the least mean square criterion. One of the significant drawbacks of Eigenface based representation is that it is highly sensitive to illumination and pose changes. To address this problem, Belhumeur *et al.* proposed Fisherface method [60] which maximizes the ratio of between-class-variance and within-class-variance. Although the Fisherface based recognition outperforms Eigenface with sufficient training samples, the efficiency of first method reduces significantly as the number of training samples decrease [61]. The problem of low training samples is addressed in [62][63], however these algorithms discard the discriminative information in null space or principle space. To fill this gap, Yang et al. proposed Complete LDA (CLDA) [64] utilizing the complete discriminative information in the null and the principle space. Lu *et al.* developed a new algorithm for CLDA with an incremental learning [65].

1.1.2 Face Recognition with Local Feature based Face Representation

Since a face image is spatially varying, significant information is not utilized by the holistic face representation. To capture the spatial structure, part based face representation methods have been proposed where different regions of face images are encoded separately. Samaria *et al.* [66] applied Hidden Markov Model (HMM) for face

recognition by dividing the face image into overlapping horizontal segments. In their method, a model is created by segmenting the face image from the top to the bottom into separate sections corresponding to forehead, eyes, nose, mouth and chin. Wiskott *et al.* [67] proposed a part based representation based on the Dynamic Link Architecture (DLA) and Gabor jets. This algorithm represents a face as labeled graphs whose nodes correspond to key points of the face such as pupils, tip of nose, corner of mouth, etc. It achieved high classification accuracy at the expense of computational complexity. Among the recent part based face recognition schemes, Local Binary Patterns (LBP) [68] has gained popularity because of its computational simplicity. LBP operator generates a series of binary codes based on the signs of pixel derivatives with respect to its neighbors. Ahonen *et al.* applied LBP to face recognition [69] by partitioning the face image into non-overlapping rectangular blocks. Dalal and Triggs [70] extended the idea of LBP to Local Ternary Pattern (LTP). LTP considers the magnitude of pixel derivatives along with its sign to generate ternary code. Zhang *et al.* proposed Local Derivation Pattern (LDP) [71], which computes the higher order derivatives of a pixel with respect to its neighbors. The local descriptors based methods (LBP, LDP, and LTP) achieve invariance to rotation and monotone transformation to certain extent and have shown quite promising results. LBP feature has been proven to be successful in the Labeled Faces in the Wild (LFW[110]) database [1][2][3]. In view of LBP's great success for face recognition, several variants are proposed later to further improve the recognition performance, such as Three-Patch LBP (TPLBP) [2], Four-Patch LBP (FPLBP) [2], Local Quantized Patterns (LQP) [4] and so on.

1.1.3 Face Recognition with Learning-based Face Representation

Although the above handcrafted features have received great success for face recognition in the past decades, two main limitations are unavoidable for them: it is difficult to define an optimal encoding method manually, since the handcrafted features are always based on our limited knowledge and experience. Therefore, there are many other methods which try to learn the discriminative feature representation automatically from images, instead of giving the specific and strait manner to obtain features.

For example, Cao et al. [5] proposed to automatically learn the local image structure encoder from training images for face recognition. Lu et al. [6] proposed an unsupervised feature learning method to learn hierarchical feature representations for face image. With the popularization of deep learning technologies, convolutional neural

networks have been widely used for face recognition. For example, Wen et al. [7] proposed a discriminative feature learning approach by using a new supervision method, center loss, to supervise the optimal process of CNNs. They achieved the state-of-the-art accuracy on several important face recognition benchmarks, such as Labeled Faces in the Wild (LFW[110]), YouTube Faces (YTF) and MegaFace Challenge. Sun et al. [8] proposed to learn a set of high-level feature representations, referred to as Deep hidden IDentity features (DeepID), for face verification. They obtained the features by taking the last hidden layer neuron activations of deep convolutional networks. This simple setting of feature extraction is proven to be successful for many severe challenges of face recognition.

1.2 Challenges and Difficulties

As it was shown above, several effective methods are proposed for discriminative face representation and robust face classification, and their effective performance is proven on various benchmarks. However, there are still many challenges and difficulties in this field. We divide the challenges and difficulties into 2 classes: noise in the face data, and noise in the labels. In the following, we give more detailed introduction into these challenges, and show how to solve these difficulties in the next subsection.

In the first class, the noises are accompanying with the face image themselves, including various deformations, illumination variations and so on. Specifically, they can be summarized into the following types:

- Noise in the data capture process: Gaussian white noise is ubiquitous and unavoidable in the image capture process, which would lead to the unavoidable presence of noise in the resulted face images. To solve this problem, we proposed a method to weaken the effect of this noise and we will introduce our idea in the following section.
- Deformations: In most of the cases, we cannot ensure the face in the input image to be in the frontal view. The faces may suffer from huge deformations in real applications. In this thesis, we provide a new face alignment method to increase the ability of the face recognition system for various views, in case of face deformations, translations, and content-noises.
- Other noise/difficulties existing in the face images: When one face is captured in a dark room, most of the pixels in the images are prone to be low values; while when the same face is captured within a bright environment, most of the pixels would go to another extreme. Different face images have various

illumination conditions, with various shading and shadows, reflection of light and so on. Besides the various illumination conditions, occlusion, aging condition, low resolution images etc., several kinds of factors would affect the face image and therefore lead to a large or small influence on the face recognition system. In this thesis, we focus on the previous challenges, but also with a gentle consideration of other types of noise including pose variations, expression changes, illuminations and so on.

In the second class, the noise mainly focuses on the labels of the face images. With the beginning of big data era, numerous unlabeled or wrong-labeled face images appeared in our life. For example, the face images on the internet are always accompanied with noisy labels.

- One major case is the existence of label flipping error, that is one face image be labeled as another wrong identity in the given label list. How to deal with these label-flipped images is a classical problem in the field of machine learning.
- One other case is the problem of outlier images. Some face images don't belong to any known identity. They should be given a new identity label, but the image capture system would give them an already known label, which would lead to noisy label samples in many face images.
- There are also cases that data has no labels, thus we propose a unsupervised multi-view face clustering method in this thesis.

1.3 Main Objectives

In this thesis, we aim at the robust face recognition method under uncontrolled environment. Face recognition methods have achieved quite promising results with large-scale dataset, especially when the dataset is clean and well-labeled. Yet, a clean large-scale dataset is quite rare except for academic use. In real-world applications, the environments are always uncontrolled, and the capturing of images may suffer from noises such as content-noise, occlusions, pose variations and so on. Besides, since labeling is quite a human-consuming work, there are also problems such as insufficient labeling, labeling noise, or even unlabeled datasets. Thus to explore face recognition methods that are robust to these kinds of problems is significant. Also, there are cases when there's only limited-scale dataset in one specific modal and large-scale dataset in other modals, thus the exploration of transfer learning methods among multiple-modals is significant. In this thesis, we seek to tackle the above-mentioned problems and propose several methods.

1.4 Contributions of the Thesis

In this thesis, we present several methods to tackle different problems.

- To alleviate the negative effect of noises and face deformations in Face Alignment and Tracking, we introduce the concept of directional features and utilize these features into the template orientation and tracking process. Through the extraction of complete information in the image, including directional and scale information, the tracking accuracy can be significantly improved. Besides, through the parallel optimization on GPU, the directional method does not increase much computational complexity.
- To tackle the problem of appearance noise in face images, including Gaussian white noise and missing pixels, we explore both directional and texture information from face images and present a shallow face recognition method. Textural and directional features are captured at the holistic and part based levels resulting in a robust face descriptor.
- To tackle the problem of unsupervised data, we propose a novel Multi-View Predictive (MVP) latent space learning model and apply it to multi-view clustering. By learning a multi-view graph with adaptive view-weight learning, MVP effectively combines the complementary information from multi-view data. Experimental results on benchmark datasets show that MVP outperforms the state-of-the-art multi-view clustering and unsupervised dimension reduction algorithms.
- Face recognition suffers from severe image blur, illumination variations, low resolution and other variations. We present how to use high-order representation of features for robust face recognition. A query sample is first modeled as a global Gaussian and then represented jointly on the dictionary of mean vectors and the dictionary of co-variance matrices. Experiments on face recognition show that the proposed CSRGC outperforms the state-of-the-art algorithms.
- To tackle the problem of label noise in datasets, we propose a Robust Deep Face Recognition (RDFR) method by an automatic outlier removal. The noisy faces are automatically recognized and removed, which can boost the performance of the learned deep models. The proposed method can ensure the recognition performance of the deep model on different datasets and reduce the training time.

- To tackle the problem of insufficient dataset for a particular task, we explored the possibility of transferring tasks on large-scale datasets to tasks on a limited-scale dataset, and a multi-task deep face recognition method is presented.

The relationship between chapters and publications are shown in figure 1.1.

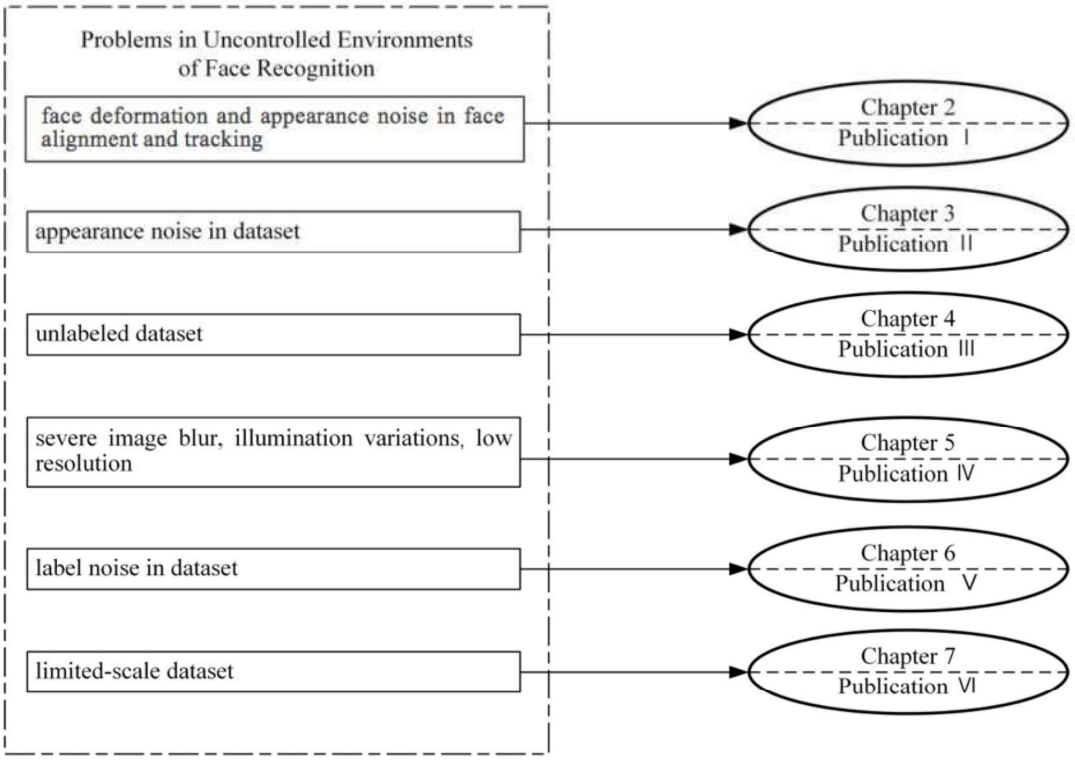


Figure 1.1. *The relationship between chapters and publications.*

1.5 Author's Contribution in the Publications

Publication I studies how directional information improves the accuracy of alignment and tracking system. The author designed the systems, wrote the paper and ran the experiments.

Publication II explores directional and textural information for face representation and proposed a robust face recognition system in noisy conditions. The author proposed the feature extraction using fixed scale filters and ran the experiments.

Publication III tackles the problem of face clustering with unsupervised data, and proposed a multi-view predictive latent space learning model and apply it to multi-view clustering and unsupervised dimension reduction. The author designed the model and wrote the paper.

Publication IV presents how to use high-order representation of features for robust face recognition. The author designed the model and wrote the paper.

Publication V studies the problem of label noise in datasets, and proposes a robust deep face recognition method with automatic outlier removal. The author designed the method and supervised writing the paper.

Publication VI studies the problem of insufficient data, and transfers tasks on large-scale datasets to tasks on a limited-scale dataset. The author designed the method and supervised writing the paper.

1.6 Organization of the Thesis

This thesis consists of an introductory part and 6 publications. The structure of the thesis is organized as follows:

After this introduction in Chapter 1, Chapter 2 introduces the concept of directional features and utilizes these features into the face alignment process, and proposes Directional Adaptive-Scale Lucas-Kanade Pyramid (DASLKP). Chapter 3 deals with the pixel noise in datasets, and proposes a face recognition method exploring both global directional and local textural information. Chapter 4 focuses on the problem of unlabeled dataset and presents a unsupervised multi-view face clustering method. Chapter 5 presents how to use high-order representation of features for robust face recognition. Chapter 6 focuses on the problem of label noise in datasets, and proposes a method to remove the data with noisy labels thus increases the performance of face recognition tasks. Chapter 7 focuses on the problem of limited-scale dataset in particular tasks and presents a multiple-task face recognition method. Finally, in Chapter 8, we conclude the thesis.

2. Exploring Directional Features for Face Alignment under Noisy Conditions

2.1 Introduction

In real applications, such as face recognition systems, the images to be aligned or tracked are always captured by surveillances, which means that the tracking objects may suffer from severe noise conditions. If we analyze the structure of an image, we may find that all images can be decomposed into points, lines, edges and textures, which are locally defined by their position, orientation and scale. These specific features encode a great proportion of information contained in images. The collection of all this information would be contributive in the orientation of particular features, particular boundaries, and thus the orientation of the object image.

In this chapter, we introduce the concept of directional features and utilize these features into the template orientation and tracking process. Instead of template tracking based on the observed noisy images, we extract directional estimates of these observations and analyze the coordinate projection of object's features and boundaries.

The rest of this chapter is arranged as follows: In Section 2.2 we give a brief review of related works concerning to Lucas-Kanade series of alignment methods. In Section 2.3, we explore directional features and how it is utilized in Lucas-Kanade affine alignment. Section 2.4 gives the analysis of parameters when exploring directional features. Section 2.5 presents the experimental results in both accuracy and complexity. We then conclude in Section 2.6.

2.2 Related Works

The alignment or registration of a pair of images is an operation required in many applications. Image alignment consists of moving, and possibly deforming, a template to minimize the difference between the template and an image [17]. One particular concern in many applications of image alignment is the efficiency of the algorithm. To efficiently align a template image to a reference image, Lucas and Kanade made a series of pioneer work by minimizing the sum of the squared difference similarity function. The earliest image alignment algorithm was the Lucas-Kanade algorithm [18]. In this algorithm, iterative parameter updates to alignment parameters are obtained by

multiplying the Jacobian with the inverse Hessian of the similarity function. Another concern of image alignment algorithms is the robustness of the algorithm. Since the base of image alignment is template matching, what if the template does not match the image because of occlusions, illumination change, and noise? To handle partial occlusions, appearance variations and presence of background pixels, robust versions of the template matching algorithm were proposed, e.g. [17][19]. The goal of the robust algorithms is to use a weighted least-squares process, such that occluded regions, background pixels and regions where brightness have changed would be considered as outliers and would be suppressed.

However, these series of LK methods ([18][20]) only provide limited robustness to noises on image and template. In such cases, there are researches focused on proposing new similarity functions. Instead of sum of squared differences used in the Lucas-Kanade framework [20][21], Mutual Information (MI) is utilized which tolerates nonlinear relationships between the intensities in image and is robust to noise, and an inverse compositional formulation for MI is developed. Then in 2009, the concept of pyramidal Lucas-Kanade method is first proposed in [22] which focuses on the application of an optical flow. Later in [23], the pyramidal method is extended to affine Lucas-Kanade feature tracking method. By extracting image pyramids from the original images and iteratively implementing LK algorithm for each level image, the LKP method gained a lot in robustness and accuracy.

Yet, in real scenario, especially for object tracking applications, the images are always captured by surveillances meaning that the object can be extremely noisy. In such cases, we first apply a powerful tool, local polynomial approximation (LPA) [24] into the LKP method and propose a novel Anisotropic Multi-Scale Lucas-Kanade Pyramid (AMSLKP) method [141], which for the first time introduces the concept of multiple directions to LK feature tracking process. The LPA method has been utilized in a number of applications related to image processing such as image denoising [25][26], image deblurring [27], image reconstruction [28], phase unwrapping [29], and color filter array interpolation [30]. In AMSLKP method, instead of calculating gradients in only one direction with a fixed scale, we utilize multiple directions with multiple scales and adaptively select the optimum scale for each pixel using Intersection of Confidence Interval (ICI) [25]. Directional estimates of noisy image are also calculated in similar manner. The final estimate and gradients are fused from the directional inputs and then utilized in the LK pyramid. The AMSLKP method gains accuracy by evaluating estimates and gradients for all level images in the pyramid. Yet the AMSLKP algorithm is complex and degrades severely as we amplify noise over a particular level.

2.3 Directional Adaptive-Scale Lucas-Kanade Pyramid (DASLKP)

In this section, we introduce the concept of directional features and utilize these features into the template orientation and tracking process, and propose Directional Adaptive-Scale Lucas-Kanade Pyramid (DASLKP). We extract directional estimates of the observations and analyze the coordinate projection of object features and boundaries. In detail,

1. First, we extract directional information from the object image at multiple scales by generating directional estimates using LPA filters. The result of applying these filters on object images and templates is a set of images which enhances the features in particular directions using kernels with different scale.
2. Further, considering the complexity of the method, ICI is used to adaptively select the optimum scale for each pixel in each direction. The result is a set of directional images and templates.
3. We use the set of directional images and templates to calculate the projection parameters. The warp parameters are calculated by minimizing the differences between all directional templates and directional images warped back onto the coordinate frame of the template. Additional accuracy gains are made through these additional directional features.

2.3.1 Generating Adaptive-Scale Directional Images (ASDIs)

Let I and T be observed 2D grey-scale image and template. The two quantities $I(\boldsymbol{x}) = I(x_0, x_1)$ and $T(\boldsymbol{x}) = T(x_0, x_1)$ are then the grey-scale values of the two images at the location $x = [x_0 \ x_1]^T$, where x_0 and x_1 are the two pixel coordinates of a generic image point x. Here, we already consider the presence of additive noise η, i.e. $I(\boldsymbol{x})$ and $T(\boldsymbol{x})$ are the observations of true images $Y_I(\boldsymbol{x})$ and $Y_T(\boldsymbol{x})$.

$$I(\boldsymbol{x}) = Y_I(\boldsymbol{x}) + \eta_I(\boldsymbol{x}) \tag{2.1}$$

$$T(\boldsymbol{x}) = Y_T(\boldsymbol{x}) + \eta_T(\boldsymbol{x}) \tag{2.2}$$

The idea here is to extract directional information from the object image at multiple scales. This information is extracted by generating directional estimates of the image.

Here we use Local Polynomial Approximation (LPA) technique [30] to obtain the directional estimates of faces at multiple scales.

LPA is applied for linear filter design using a polynomial fit in a sliding window. It is based on nonparametric estimation of the signal. The signal can be represented as a linear combination of polynomials centered on the observation coordinates using the Taylor series expansion. The method for generating the LPA filters is discussed in details in [30]. The kernel g_{h,θ_i} for direction θ_i and scale h is generated as:

$$g_{h,\theta_i}(\boldsymbol{x},\boldsymbol{X}_s) = w_{h,\theta_i}(\boldsymbol{x}-\boldsymbol{X}_s)\boldsymbol{\phi}_h^T(\boldsymbol{x}-\boldsymbol{X}_s)\Phi_h^{-1}\boldsymbol{\phi}_h(\mathbf{0}) \tag{2.3}$$

$$\Phi_h = \sum_s w_{h,\theta_i}(\boldsymbol{x}-\boldsymbol{X}_s)\boldsymbol{\phi}_h(\boldsymbol{x}-\boldsymbol{X}_s)\boldsymbol{\phi}_h^T(\boldsymbol{x}-\boldsymbol{X}_s) \tag{2.4}$$

$$\phi_h(\boldsymbol{x}) = \frac{(-1)^{|\boldsymbol{k}|}\boldsymbol{x}^{\boldsymbol{k}}}{\boldsymbol{k}!},\ \boldsymbol{k}\in Z^2:\ \forall|\boldsymbol{k}| \le m \tag{2.5}$$

where $w_{h,\theta_i}(\boldsymbol{x},\boldsymbol{X}_s)$ is directional window in direction θ_i with scale size h, m is the order of polynomial, $\boldsymbol{\phi}_h \in R^M$ is the complete set of linearly independent 2D polynomials of the powers from 0 till m, and the length of the vector is equal to:

$$M = (m+2)!/2\cdot m! = (m+2)(m+1)/2 \tag{2.6}$$

An important factor in the design of the LPA kernels is the choice of directions $\{\theta_i\}$ and scale sizes $\{h\}$. The directions should be chosen such that the selected directions are aligned with the majority of the prominent object features and there is no redundancy in them. The scale determines the number of neighbouring pixels considered in calculating the directional estimates at a point. Thus, in case of noisy image model, it is better to use a considerable number of scales so that a strong estimate is obtained, however, if we assume noise to be absent then the smallest scale itself is enough. A detailed discussion for selecting these parameters is done in Section 2.4. The object images are convolved with these filters to obtain the directional estimates at multiple scales. For the object image $I(x)$ the directional estimates for direction θ_i and scale h are given by

$$\hat{I}_{h,\theta_i}(\boldsymbol{x}) = I(\boldsymbol{x}) \circledast g_{h,\theta_i}(\boldsymbol{x}) \tag{2.7}$$

Here $\circledast$ indicates the convolution operator.

In the above design of the directional LPA kernels, the kernels are specially constructed for each desirable direction θ_i and for each scale $h \in H$. In this way we obtain a set of the kernels $g_{h,\theta_i}(\boldsymbol{x},\boldsymbol{X}_s)$ which can be treated as a filter bank of the directional LPA.

Then, considering the complexity of this method, Intersection of Confidence Interval (ICI) [25] is used to adaptively select the optimum scale for each pixel in each direction. The idea behind ICI is that the estimate of every point depends on its neighbourhood, if an estimate is evaluated using a small neighbourhood it would have small bias but high random error, on the other hand, if a very large neighbourhood is considered for the estimate, it would have a high bias. ICI finds the neighbourhood which strikes a proper balance between the bias and randomness. Since neighbourhood in LPA filters is represented by scale, ICI algorithm finds the largest scale for which the deviation of the estimate from the smallest scale is not too large. It uses a sequence of confidence intervals, for each scale, in all directions. The confidence interval, $Q_j = [L_j, U_j]$, for scale h_j gives the upper bound U_j and lower bound L_j between which the estimates is expected to lie. For a scale h_j, the possible intersection at point x is given as:

$$Q_j = [L_j, U_j]$$

$$\text{where } U_j = \hat{I}_{\theta_i,h_j}(x) + \Gamma \cdot \sigma_{\hat{I}_{\theta_i,h_j}}(x) \tag{2.8}$$

$$L_j = \hat{I}_{\theta_i,h_j}(x) - \Gamma \cdot \sigma_{\hat{I}_{\theta_i,h_j}}(x)$$

Where Γ is the threshold which plays an important role (it has to be tuned to set the width of the confidence interval), $\hat{I}_{\theta_i,h_j}(\boldsymbol{x})$ are the first order estimates obtained in equation (2.7) and $\sigma_{\hat{I}_{\theta_i,h_j}}{}^2 = \sigma^2 \sum g_{\theta_i,h_j}(\boldsymbol{x})$ is the standard deviation of the additive noise modeled to be present in the image. The estimation of σ is independent of ICI method; in our method we convolved the original image with Daubechies kernel and took the median of the result as the estimate.

The algorithm starts from the smallest scale and moves toward the large scales, and for each scale it computes the confidence interval as mentioned in equation (2.8). For a given direction θ_i, we can generate all the estimates $\left\{\hat{I}_{\theta_i,h_j}\right\}_{h_j \in H}$ for all scales $h_j \in H$. Then ICI rule is utilized in order to find the point-wise optimum scale $h^+ = h_{j_+}$ for all directions $\{\theta_i \in D\}$ and thus the directional estimate $\hat{I}_{\theta_i} = \hat{I}_{\theta_i,h_+}$.

Let $\bar{L}_{j+1} = max\{\bar{L}_j, L_{j+1}\}$, $\underline{U_{j+1}} = max\{\underline{U_j}, U_{j+1}\}$. The basic idea of ICI is to find the largest j when $\bar{L}_j \leq \underline{U_j}|_{j=1,2,\dots,J}$ is still satisfied. Denote this largest value j_+.

Then this j_+ is the largest of those j for which the confidence intervals Q_j have a point in common and the ICI adaptive scale is $h^+ = h_{j_+}$. This scale is at last used for the directional estimation of that pixel. After applying ICI, an image is obtained for each direction. The face images obtained after ICI are henceforth called Adaptive-Scale Directional Images (ASDIs). Taking 4 directions for instance, the optimized directional faces are shown in figure 2.1. In such a way, the directional estimates $\{\hat{I}_{\theta_i}\}_{\theta_i \in D}$ of the observed image can be calculated using equation (2.9).

$$\hat{I}_{\theta_i}(\boldsymbol{x}) = \hat{I}_{\theta_i,h}(\boldsymbol{x})|_{h = h_+(\boldsymbol{x}, \theta_i)} \tag{2.9}$$

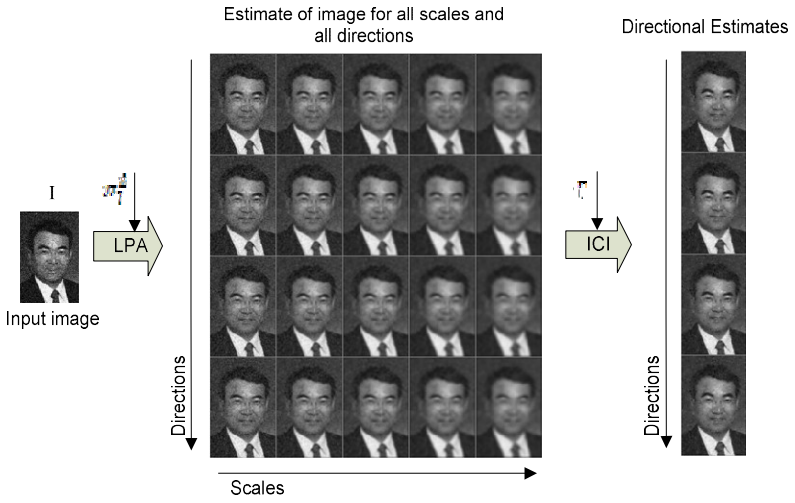


Figure 2.1. *The process of generating optimized directional images is shown. The image is first convolved with the set of directional LPA filters, generated for five scales and four directions. Then, for each direction, the optimum scale is adaptively selected using ICI rule resulting in a Directional Estimate. Here, different directional estimates highlight different directional features.*

2.3.2 Directional Lucas-Kanade Algorithm

Let us first recall the iterative process of Lucas-Kanade algorithm: LK tracking is to align the template $T(\boldsymbol{x}) = T(x_0, x_1)$ to the input image $I(\boldsymbol{x}) = I(x_0, x_1)$. Let $\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p})$ denote the parameterized set of allowed warps, where $\boldsymbol{p} = [p_1, \ldots p_n]^T$. The warp $\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p})$ takes the pixel $\boldsymbol{x}$ in the coordinate frame of the template T and maps it to the sub-pixel location $\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p})$ in the coordinate frame of the image I. The goal of traditional LK affine tracking [20] is to find the parameters $\boldsymbol{p}$ such that $T(x)$ and

$I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))$ are 'similar'.

In general, the number of warp parameters can be arbitrarily large and $W(x;p)$ can be arbitrarily complex. The vector of parameters would be $\boldsymbol{p} = [p_1\ p_2\ p_3\ p_4\ p_5\ p_6]^T$, with the parameterized set of affine warp as following:

$$\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ 1 \end{pmatrix} \tag{2.10}$$

Given directional estimates $\{\hat{T}_\theta\}_{\theta\in D}$ and $\{\hat{I}_\theta\}_{\theta\in D}$, we aim to minimize the sum of squared errors between all the templates $\{\hat{T}_\theta\}$ and images $\{\hat{I}_\theta\}$ warped back onto the coordinate frame of the templates:

$$\epsilon(p_1,p_2,p_3,p_4,p_5,p_6) = \sum_{\theta\in D}\sum_x \left(\hat{T}_\theta(\boldsymbol{x}) - \hat{I}_\theta(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))\right)^2 \tag{2.11}$$

To optimize the expression in equation (2.11), the Lucas-Kanade algorithm assumes that a current estimate of $\boldsymbol{p}$ is known and then iteratively solves for increments to the parameters $\Delta\boldsymbol{p}$; i.e. the following expression is minimized:

$$\sum_{\theta\in D}\sum_x \left(\hat{T}_\theta(\boldsymbol{x}) - \hat{I}_\theta(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p}))\right)^2 \tag{2.12}$$

with respect to $\Delta\boldsymbol{p}$, and then the parameters are updated:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p} \tag{2.13}$$

To the equation (2.13), we apply a first order Taylor expansion on $\hat{I}_\theta(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p}))$ to give:

$$\sum_{\theta\in D}\sum_{\boldsymbol{x}} \left[\hat{I}_\theta(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) + \nabla\hat{I}_\theta \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} - \hat{T}_\theta(\boldsymbol{x})\right]^2 \tag{2.14}$$

where $\nabla\hat{I}_\theta$ is the gradient of the image $\hat{I}_\theta$ evaluated at $W(x;p)$. A partial derivative with respect to $\Delta\boldsymbol{p}$ is then obtained:

$$2\sum_{\theta\in D}\sum_{\boldsymbol{x}} \left[\nabla\hat{I}_\theta \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[\hat{I}_\theta(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) + \nabla\hat{I}_\theta \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} - \hat{T}_\theta(\boldsymbol{x})\right] \tag{2.15}$$

Assuming a locally parabolic shape and setting the partial derivative to zero gives a closed form solution for updating $\boldsymbol{p}$, which minimizes equation (2.16):

$$\Delta\boldsymbol{p} = H_{\bar{D}}^{-1}\sum_{\theta\in D}\sum_{\boldsymbol{x}} \left[\nabla\hat{I}_\theta \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[\hat{T}_\theta(\boldsymbol{x}) - \hat{I}_\theta(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))\right] \tag{2.16}$$

$$H_D = \sum_{\theta \in D} \sum_{x} \left[\nabla \hat{I}_\theta \frac{\partial W}{\partial p} \right]^T \left[\nabla \hat{I}_\theta \frac{\partial W}{\partial p} \right] \tag{2.17}$$

Here we successfully extract the directional information from both the tracking image and the template, as described by $\{\hat{I}_\theta\}_{\theta \in D}$ and $\{\hat{T}_\theta\}_{\theta \in D}$. Now, instead of template tracking based on single observed images, we will take all these directional images and analyze the coordination projection in the tracking process. Given a specific level l, we have generated the directional images $\{\hat{I}_\theta^l\}_{\theta \in D}$ and directional templates $\{\hat{T}_\theta^l\}_{\theta \in D}$.

2.3.3 Directional Lucas-Kanade Pyramidal Algorithm

We extend the Directional LK discussed in previous section based on pyramids. First, let us give the definition of pyramid of a generic image I of size $n_x \times n_y$. Let $I^0 = I$ be the zeroth level image. The image is the highest resolution image (raw image). The image size at zeroth level is defined as $n_x^0 = n_x,\ n_y^0 = n_y$. The pyramid representation is then built in a recursive fashion: compute I^1 from I^0, then I^2 from I^1, and so on… Let $L = 1, 2, \ldots$ be a generic pyramidal level, and let I^{l-1} be the image at level $l-1$. Denote n_x^{l-1}, n_y^{l-1} the width and height of I^{l-1}. Then [23]

$$\begin{aligned} I^l(x,y) = {} & \frac{1}{4} I^{l-1}(2x, 2y) \\ & + \frac{1}{8}\Big(I^{l-1}(2x-1, 2y) + I^{l-1}(2x-1, 2y) + I^{l-1}(2x+1, 2y) \\ & + I^{l-1}(2x, 2y+1) \Big) \\ & + \frac{1}{16}\Big(I^{l-1}(2x-1, 2y-1) + I^{l-1}(2x+1, 2y+1) \\ & + I^{l-1}(2x+1, 2y-1) + I^{l-1}(2x-1, 2y+1) \Big) \end{aligned} \tag{2.18}$$

The width n_x^l and height n_y^l of I^l are the largest integers that satisfy the conditions:

$$n_x^l \leq \frac{n_x^{l-1}+1}{2}, \quad n_y^l \leq \frac{n_y^{l-1}+1}{2} \tag{2.19}$$

The pyramidal images $\{\hat{I}_\theta^l\}_{\theta_i \in D, l=0,1,\ldots L_m}$ and $\{\hat{T}_\theta^l\}_{\theta_i \in D, l=0,1,\ldots L_m}$ are constructed recursively using equation (2.18).

The warp parameter must be iteratively computed and updated until the variation in parameters or function values becomes sufficiently small.

From pyramidal view, the warp parameter $\boldsymbol{p}$ is also the warp parameter for the 0^{th} level image $\hat{I}^0$ i.e. $\boldsymbol{p} = \boldsymbol{p}^0$, where $\boldsymbol{p}^l$ is evaluated by minimizing the difference between the l^{th} level warped directional estimate $\hat{I}^l\left(\boldsymbol{W}(\boldsymbol{p}^l + \Delta\boldsymbol{p}^l)\right)$ and the template image T, i.e.

$$\boldsymbol{p}^l = \min_{\Delta\boldsymbol{p}^L}\left\{\sum_\theta \sum_{\boldsymbol{x}} \left(\hat{T}_\theta^l\left(\boldsymbol{W}(\boldsymbol{p}^l + \Delta\boldsymbol{p}^l)\right) - \hat{I}_\theta^l\right)^2\right\} \tag{2.20}$$

Thus we have

$$\Delta\boldsymbol{p}^l = \left(H_D^l\right)^{-1} \sum_\theta \sum_{\boldsymbol{x}} \left[\nabla \hat{I}_\theta^l \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[\hat{T}_\theta^l(\boldsymbol{x}) - \hat{I}_\theta^l\left(\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p}^l)\right)\right] \tag{2.21}$$

$$H_D^l = \sum_\theta \sum_{\boldsymbol{x}} \left[\nabla \hat{I}_\theta^l \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[\nabla \hat{I}_\theta^l \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right] \tag{2.22}$$

$\boldsymbol{p}^l$ is calculated by iteratively updating parameters:

$$\boldsymbol{p}^l \leftarrow \boldsymbol{p}^l + \Delta\boldsymbol{p}^l \tag{2.23}$$

The parameter update process is recursively implemented from ${L_m}^{\text{th}}$ level down to 0^{th} level by evaluating the initial guess for $(l-1)^{\text{th}}$ level parameter from l^{th} level. Note that in affine warp, the first 4 parameters $[p_1\ p_2\ p_3\ p_4]^T$ are related to transformation while the last 2 parameters, $[p_5\ p_6]^T$ are related to translation. Those who contribute to transformation will be the same for all levels in the pyramidal representation, while those for translation will change.

$$\boldsymbol{p}_{init}^{l-1} = [{p_1}^l\ {p_2}^l\ {p_3}^l\ {p_4}^l \quad {p_5}^l \times 2 \quad {p_6}^l \times 2] \tag{2.24}$$

2.4 Parameter Analysis

In this section, we evaluate the effect of various parameters in our experiments. Equation (2.3) shows that the value of LPA kernels g_{h,θ_i} is defined by several parameters including scale size $h \in H$, division of support $\theta_i \in D$, distribution of the window $w_{h,\theta}$, and symmetry feature of the window $w_{h,\theta}$. Equation (2.8) reveals the role of the threshold parameter Γ in ensuring the fidelity of the adaptive estimate $\hat{I}_{\theta_i,h_+}$ since the Γ value will directly influence the choice of adaptive scales.

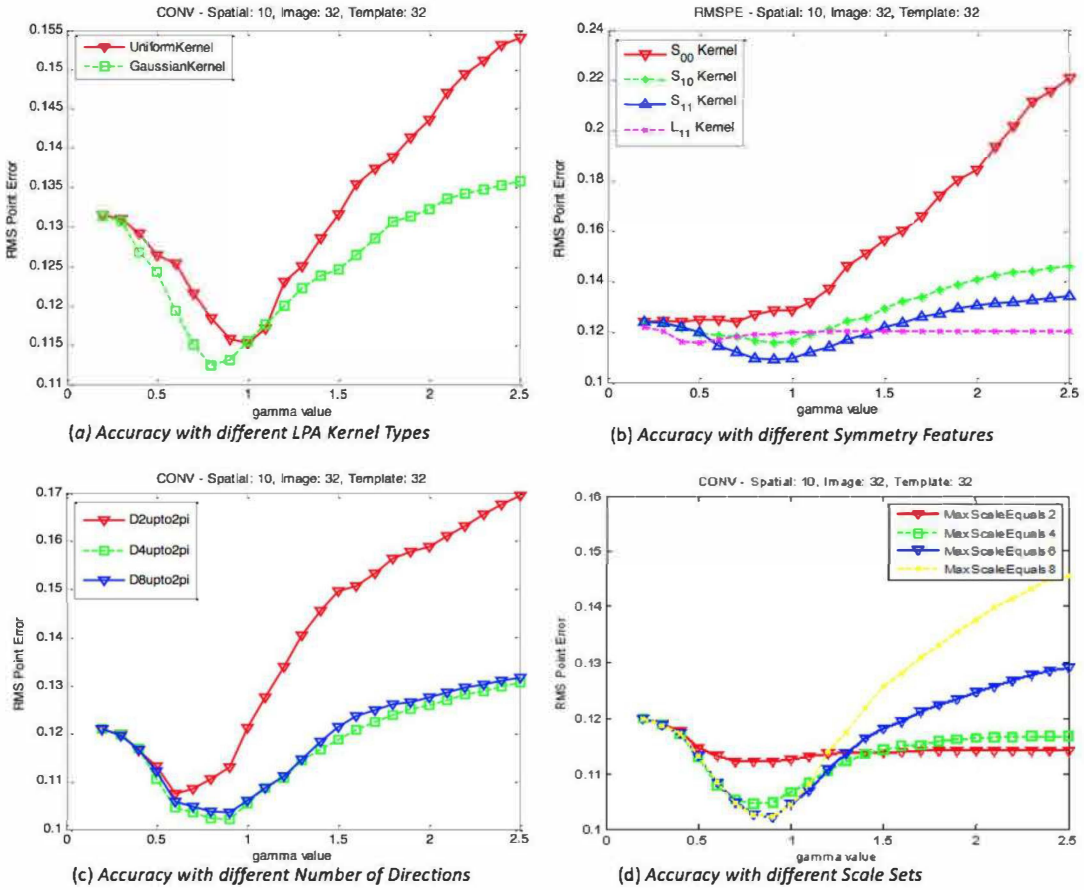


Figure 2.2. *Tracking accuracy in terms of RMS Point Error for different LPA kernels. (a) Tracking accuracy in terms of RMS Point Error for Gaussian and Uniform kernels. (b) Tracking accuracy in terms of RMS Point Error using Gaussian LPA kernels with different symmetry features. (c) Tracking accuracy in terms of RMS Point Error for kernel supports divided by different number of directions. (d) Tracking accuracy in terms of RMS Point Error for different scale sets.*

Considering their important role in the estimation of objects, the influences of all above mentioned parameters on the tracking accuracy should be evaluated. Since 6 parameters in the affine warp have different units, we use the following error measurement method rather than the errors in the parameters. Given the current estimate of the warp, we compute the destination of 3 canonical points and compare them with the correct locations. We compute the Root Mean Square Error over the 3 points of the distance between their current and correct locations. In order to yield more precise results, 100 tests are implemented for each noise condition, and the overall result for a given noise condition is averaged from the 100 test results.

The parameters can be categorized into two classes: those defining the LPA kernels, and threshold parameter Γ defining the ICI selection. Specifically, LPA parameters includes scale size $h \in H$, division of support $\theta_i \in D$, distribution of the window $w_{h,\theta}$,

and symmetry feature of the window $w_{h,\theta}$. The performance of LPA parameters changes as the parameter Γ varies. Thus in following sections, we will analyze the accuracy performances of all these LPA parameters over a given Γ range. We adjust the structure of LPA kernels by defining different windows $\{w\}$. We also manipulate the division of LPA support $\theta_i \in D$ by tuning number of directions, and extend the scale range $H = \{1, 2, \dots h_{max}\}$ by varying the maximum scale h_{max}.

In all these tests, 3 levels are utilized in the pyramid generation process, i.e. $L_m = 2$, while in each level, 5 iterations are adopted when estimating parameters. Besides, the polynomial order m used for generating LPA kernels in equation (2.6) is set to $m = [1, 0]$, meaning that totally $M = 3$, $\phi_h = [\phi_0, \phi_1, \phi_2]$, with $\phi_0 = 0$, $\phi_1 = x_1$, $\phi_2 = x_2$. And the noise imposed to image and template is zero-means Gaussian additive noise with the same standard deviation value $\sigma_I = \sigma_T = 32$. And the perturbation noise variance equals 10: $\sigma_s = 10$

2.4.1 Kernel Types

Equation (2.3) shows that different distribution of windows w_{h,θ_i} will directly influence the value of LPA kernels g_{h,θ_i} and thus affect the estimation and smoothness of observed images, resulting in different tracking accuracies. Here, we choose two types of windows: square uniform window and Gaussian window. The uniform square window means that w has equal values inside of the square support, while the Gaussian window means that w is the 2D standard Gaussian density. As a result, the Gaussian kernels g and their frequency characteristics are different from the ones considered for the uniform window mainly by their smoothness [24]. It is assumed that the linear finite-difference structure of the smoothness estimates is more appealing for Gaussian kernels than uniform kernels. Hence, in this section we will discuss the results using both square uniform and Gaussian window filters of the order $m = [1, 0]$ and of the maximum support size $h_{max} = 6$. The test is conducted with following conditions: 4 directions from $D = \left[0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right]$ are used for sector division.

Figure 2.2 (a) shows the tracking accuracy for directional LPA kernels using both uniform and Gaussian windows, the directions used here is $\theta_i \in D = \left[0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right]$. As we can see, the results are quite promising in that Gaussian windows yield better accuracy. Thus in following tests, we will leave uniform kernels out of this work and

use Gaussian distributions in the kernel generation process equation (2.3).

We assume that all kernels are finite-defined on square supports, and the symmetry feature of the supports is another consideration that may affect the tracking accuracy. The support can be symmetric, partly-symmetric, and non-symmetric with respect to the central point $(0,0)$ of LPA. In detail, considering the symmetry of the sectorial window support w, we elongate the directional basic kernel along the axis x_1, and differentiate the three types of kernels in following ways:

1. Sectorial Kernel Type 00, labeled as S_{00}: Kernels with the support even with respect to both variables x_1 and x_2, and $w(x_1,x_2) = w(-x_1,x_2)$, $w(x_1,x_2) = w(x_1,-x_2)$.
2. Sectorial Kernel Type 10, labeled as S_{10}: Kernels with the support even with respect to the variable x_2, and $w(x_1,x_2) = w(x_1,-x_2)$.
3. Sectorial Kernel Type 11, labeled as S_{11}: Kernels with the support with non-negative $x_1,\ x_2 \geq 0$.

For a given direction θ and given size of scale h, the 2D Gaussian window function w for a basic sectorial kernel of these three types are formularized as:

1. S_{00}: $w_{h,\theta} = \frac{1}{\sqrt{2\pi}} \cdot e^{\frac{-({x_1}^2+{x_2}^2)}{2}}, \quad if \left|\frac{x_1}{\sqrt{{x_1}^2+{x_2}^2}}\right| \leq sin\,\theta, \|x\| \leq h$
2. S_{10}: $w_{h,\theta} = \frac{1}{\sqrt{2\pi}} \cdot e^{\frac{-({x_1}^2+{x_2}^2)}{2}}, \quad if \left|\frac{x_1}{\sqrt{{x_1}^2+{x_2}^2}}\right| \leq sin\,\theta, \|x\| \leq h,\ x_1 \geq 0$
3. S_{11}: $w_{h,\theta} = \frac{1}{\sqrt{2\pi}} \cdot e^{\frac{-({x_1}^2+{x_2}^2)}{2}}, \quad if \left|\frac{x_1}{\sqrt{{x_1}^2+{x_2}^2}}\right| \leq sin\,\theta, \|x\| \leq h,\ x_1 \geq 0,\ x_2 \geq 0$

Other than the sectorial kernels, line-wise kernels should also be considered, labeled as L_{11}, we only give one example of line-kernel for Type 11 when $\theta = 0$:

$$w_{h,\theta} = \frac{1}{\sqrt{2\pi}} \cdot e^{\frac{-({x_1}^2+{x_2}^2)}{2}}, \quad if\ 0 \leq x_1 \leq h,\ x_2 = 0.$$

Taking 4 directions $\theta \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, maximum scale size $h = 2$ for instance, the supports for all the above mentioned four types of kernels are shown in figure 2.3.

To give a clearer analysis of how different kernel types may influence the accuracy of the proposed DASLKP method, in all the tests we first fix the scale set $H = \{1,2,3,4\}, i.e.\ h_{max} = 4$ and the direction set $D = \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, and then we evaluate the proposed DASLKP method using the four kernels described in figure 2.3. The evaluation process is conducted for variance noise conditions with respect to a wide Γ range from 0.2 to 2.5 with step-size equaling 0.1. Performance results show that non-

symmetric sectorial kernels provide more accurate object tracking. In figure 2.2 (b), the influence of different kernels together with the effect of threshold parameter Γ is also presented for noise condition: $\sigma_s = 10, \sigma_I = 32, \sigma_T = 32$. The results are supportive for S_{11} type of kernel, which is non-symmetric sectorial kernel.

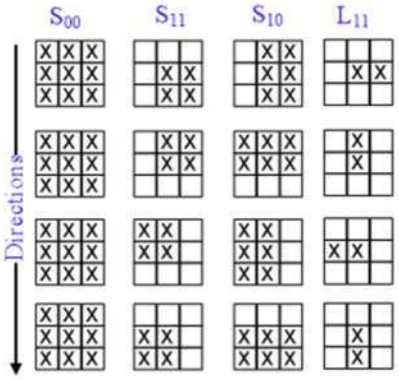


Figure 2.3. *Structure of different kernel types in the case of 4 directions with scale size equaling 2.*

2.4.2 Number of Directions

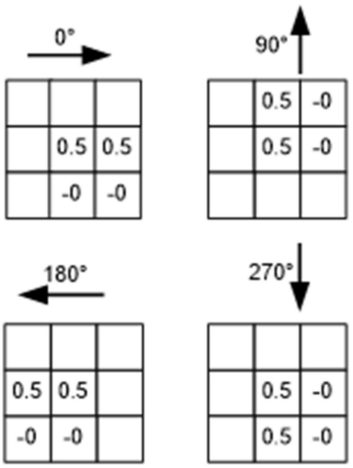


Figure 2.4. *LPA directional kernels for 4 directions and scale size 2.*

Other than kernel types, the sector division of local polynomial kernels may also influence the accuracy performance. As discussed in Section 2, estimates in different directions cover diverse information of an object image. Figure 2.4 gives an example set of LPA kernels $g_{h,\theta_i}(x, X_s)$ generated for 4 directions $\theta_i \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$ at scale $h = 2$. It can be seen from the figure that those directions correspond to horizontal right, horizontal left, vertical up and vertical down.

Recall that, to generate the directional kernel for a particular direction θ, we need to rotate the kernel support in order to direct it to the desired direction θ. The paper [24] describes in detail about the support rotation. Equation (2.4) shows that, in order to generate the LPA kernels, the summation over the support grid $\{\boldsymbol{X}_s\}$ of the observations is used. And the number of nodes used in the calculations of those kernels depends on direction θ and scale size h, and in this way the kernels can be quite different for different directions and different scales. The influence of scales will be further discussed in Section 2.4.3. In this section, we mainly focus on how the division of kernel support and directions affects the accuracy performance of object tracking.

We use LPA kernels of the order $m = [1, 0]$ obtained for the Gaussian S_{11} window with the scales $H = \{1, 2, 3, 4\}$, and then evaluate the proposed method when the LPA support is divided into 2, 4, and 8 directions, i.e. $\theta_i \in \{0, \pi\}$, $\left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, and $\left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\right\}$. Figure 2.2 (c) presents the RMSPE performance over the same Γ range from 0.2 to 2.5. The results show that best result is provided by dividing the sectors in four directions $\theta_i \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$.

2.4.3 Different Scales

The scale h defines the size of the window, which is also the number of neighboring nodes used to filter the image. The idea of varying scale and ICI rule allows a unique best scale in a point-wise manner for each point in every direction, thus improves the point-wise accuracy of estimation. Different selection of scale set H results in different supports used for estimation and thus affects the accuracy of our proposed method.

The threshold parameter Γ plays a relevant role in the selection of the adaptive scale. From equation (2.8) we observe that a too large Γ makes the inter-section of the confidence intervals more likely to be non-empty [32], and that as a consequence the chosen adaptive scale may be rather large. Analogously, with a small Γ, the empty intersection is likely to happen at the smaller scale, while for a large Γ, ICI tends to select larger scales. And too large or too small Γ results in over-smoothing or under-smoothing estimates. So theoretically, the tracking accuracy should first increase as Γ enlarges, and then degrade after a certain point. The larger h_{max} is, the more curved the plot should be. And if we extend the scale set by tuning the h_{max} value, the proposed method should intensively degrade for large Γ values as h_{max} increases.

Figure 2.2 (d) shows promising results as h_{max} increases from 2 to 8 with step-size equaling 2. We all know that the wider scale set H is, the more complicated the method is. So considering both complexity and accuracy, the optimal scale set can be found when h_{max} equals 4.

2.4.4 Γ for ICI

Threshold parameter Γ plays a relevant role in the selection of the adaptive scale. It determines the threshold of the confidence interval. The width of the confidence interval for scale h and direction θ is given by $2 \cdot \Gamma \cdot \sigma_{\hat{I}_{h,\theta}}$, which depends only on Γ and noise estimate $\sigma_{\hat{I}_{h,\theta}}$ of the image. Analogously, with a small Γ, the empty intersection is likely to happen at the smaller scale, while for a large Γ, ICI tends to select larger scales. Too large or too small Γ results in over-smoothing or under-smoothing estimates. So theoretically, the tracking accuracy should first increase as Γ enlarges, and then degrade after a certain point. The optimum value of Γ depends on the amount of noise present in the image. If the noise is not present then smaller Γ values give good results, but if noise is present then larger Γ values are required. Here we study the effect of noise on the optimum value of Γ. In following tests, optimum settings from all above sections are used, i.e. LPA kernels of the order $m = [1, 0]$ obtained for the Gaussian S_{11} window with the scales $H = \{1, 2, 3, 4\}$ and the support divided into 4 directions $\theta_i \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$. The value of Γ varies from 0.2 to 2.5 with step-size equaling 0.1, and zero-means Gaussian additive noise is considered. In this test, we assume that both template and image have the same standard deviation value: $\sigma_I = \sigma_T \in \{0, 8, 16, 24, 32\}$.

Figure 2.5 shows how noise level influences the optimum gamma selection. The five bar-figures give how the accuracy performance varies in terms of RMS Point Error as we increase noise standard deviation σ from 0 up to 32. The optimum Γ values are high-lighted by purple bars. And the optimum Γ increases from 0.5 to 0.9 as we enlarge the noise level from 0 to 32. Note that, for non-noisy case, i.e. $\sigma = 0$, the RMSPE value is approaching zero and the accuracy difference among the Γ values from the range $[0.2 \rightarrow 0.8]$ is trivial.

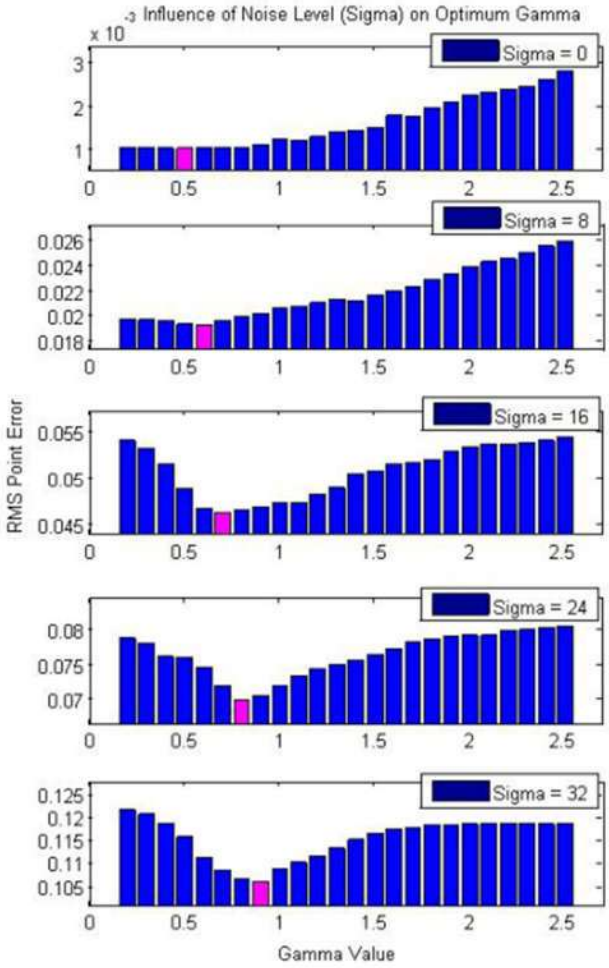


Figure 2.5. *Influence of noise level σ on the choice of Optimum Γ.*

2.5 Experimental Results

To demonstrate the robustness of the proposed algorithm, we present some evaluations and comparisons on facial objects. LKP method and AMSLKP method are used for performance comparison.

The observed image I and template T may be degraded by the addition of Gaussian white noise with a range of standard deviations. The experiments are conducted over 3 different scenarios depending on noise conditions of the objects; here, we label these three scenarios as Scenario_1, Scenario_2 and Scenario_3:

1. Scenario_1: Non-noisy for both image and template;
2. Scenario_2: Image I is noisy while template T is not;
3. Scenario_3: Image I and template T are both noisy.

Moreover, for all these 3 scenarios, various noise intensities are also considered. To parametrize, the standard deviation σ of the added noise is chosen from the set

$\sigma_I, \sigma_T \in \{0, 8, 16, 24, 32, 64\}$, and the perturbation noise variance σ_s is chosen from set: $\sigma_s \in \{2, 4, 6, 8, 10\}$.

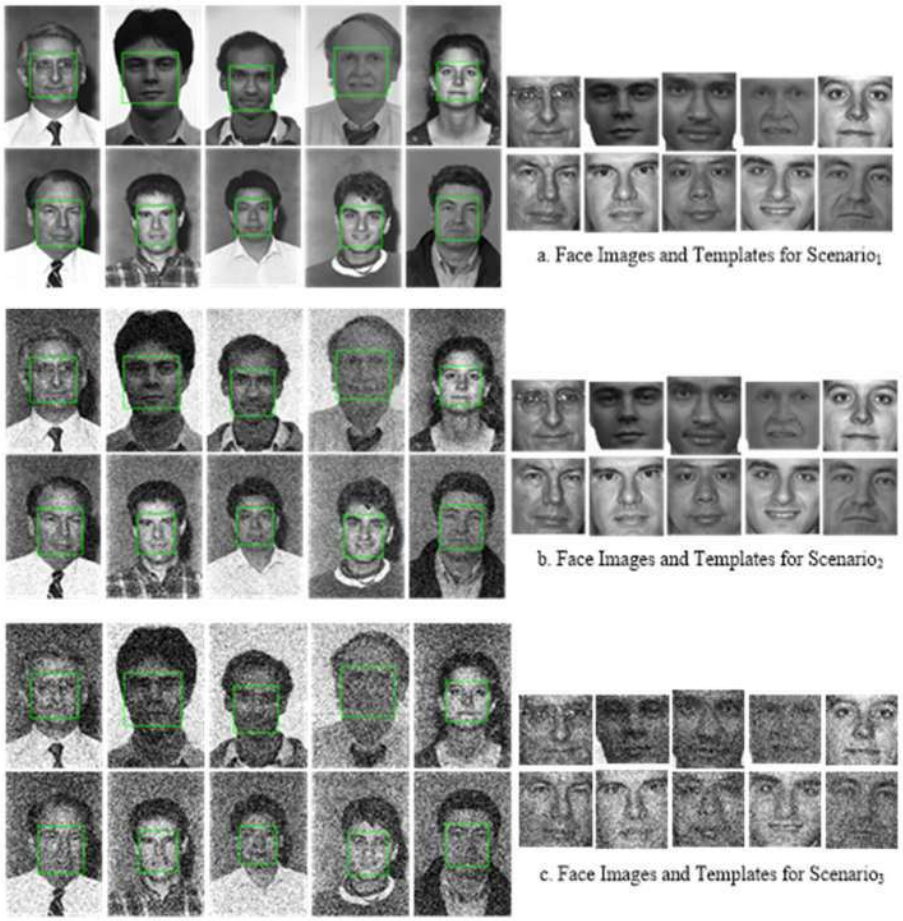


Figure 2.6. *FERET Face Images and Templates generated using identity warp for all three scenarios.*

As we already described in the beginning of Section 2.4, same RMS Point Error (RMSPE) measurement over the 3 canonical points is utilized in the robustness evaluation process. Overall result for a given noise condition is gained through averaging the RMSPE results of 100 such tests.

In all these tests, 3 levels are utilized in the pyramid generation process, i.e. $L_m = 2$, while in each level, 5 iterations are adopted when estimating parameters. Besides, we use LPA kernels of the order $m = [1, 0]$ obtained for the Gaussian S_{11} window with the scales $H = \{1, 2, 3, 4, 5, 6\}$, and the support divided into 4 directions $\theta_i \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$. As discussed in Section 2.4.4, the value of optimum Γ will be chosen from $[0.2 \rightarrow 1.0]$ according to different noise conditions.

One possible application of this tracking method is to align and track face data for face

recognition systems, we evaluate this proposed method over several faces chosen from FERET dataset [138][139]. In order to give a more general and comprehensive analysis of our proposed method, totally we choose 10 frontal faces from different races and different genders and label them as I. For each face, we manually select the central part of the face as template T with identity warp, labeled by the green rectangle area in figure 2.6 (a).

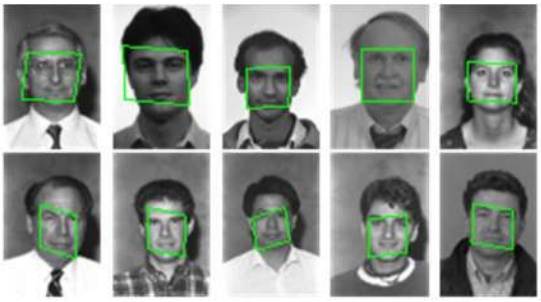


Figure 2.7. *FERET Face Images and Templates to be aligned.*

As we have described in above section, additive Gaussian noises with various intensity would be imposed to the images and templates. Besides, with the perturbation noise variance σ_s, we randomly perturb the canonical points with additive white Gaussian noise and fit for the affine warp parameters $\boldsymbol{p}$ that these perturbed points define. We then use identity warp as the initial parameters and run LKP, AMSLKP, and proposed DASLKP alignment methods.

2.5.1 Perturbation Noise

Table 2.1. RMSPE results for none appearance noise cases

Γ	σ_I	σ_T	σ_s	RMSPE		
				LKP	AMSLKP	DASLKP
0.2	0	0	2	0.04711	0.0036699	0.00067136
0.2	0	0	4	0.049057	0.0031997	0.00052876
0.2	0	0	6	0.050876	0.0032011	0.00065578
0.2	0	0	8	0.052934	0.0035188	0.00092312
0.2	0	0	10	0.054407	0.0044801	0.0010176

For none appearance noise case, i.e. $\sigma_I = \sigma_T = 0$, we randomly perturbed the canonical points and then generate following templates. Figure 2.7 shows one case

when $\sigma_s = 10$, as labeled by green rectangles. And then we start to align these templates from identity warp positions, as shown in figure 2.6 (a). The RMSPE results for various conditions $\sigma_s \in \{2, 4, 6, 8, 10\}$ are shown in Table 2.1.

As we can see from Table 2.1, our proposed method gains another 80% percent over AMSLKP work and the alignment error is approaching zero.

2.5.2 Gaussian Noise

In this scenario, the images are degraded by Gaussian noises. We show the case when $\sigma_I = 32$ in figure 2.6 (b). Templates are generated in the way same as Scenario$_1$. The RMSPE results for perturbation noise conditions $\sigma_s = 10$ are shown in Table 2.2. And the proposed method surpasses other methods in both accuracy and stability. Especially when severe noise level is imposed, our method gains around 50% over LKP method while AMSLKP method diverges.

Table 2.2. RMSPE results for Gaussian noise cases

Γ	σ_I	σ_T	σ_s	RMSPE		
				LKP	AMSLKP	DASLKP
0.6	8	0	10	0.05727	0.017909	0.01338
0.7	16	0	10	0.064365	0.048944	0.028478
0.8	24	0	10	0.082403	0.067247	0.04766
0.9	32	0	10	0.1059	0.081086	0.06369
1.2	64	0	10	0.36301	Diverged	0.18842

2.5.3 Pose Variations

To evaluate the effectiveness of DASLKP method in real scenario, we built a small dataset that contains images taken from several persons with pose variations. The dataset is shown in figure 2.8 (a), the green rectangle is the ground-truth central area. For each person, we take the central area of one image (the left column) as template, and utilize DASLKP method to locate the central area of other images with the template. Figure 2.8 (b) gives the initial points that the template-matching starts from. For each person, the 2nd column uses same image with deformed initial rectangles, while the latter columns use different images with pose variations. Table 2.3 gives the matching

accuracy results using the proposed method with the above-mentioned parameter settings. Results show that, if only perturbation noises are opposed on initial points, the person can be located correctly on the same image (2nd column), yet for other images of the same person (other columns), our method diverges. This is because DASLKP method is content-based, directional information provides additional robustness only to transformations, translations as well as noises, yet it cannot work well when content is somehow lost or changed in case of pose variations. In latter chapters, we will introduce deep methods that can alleviate pose variation problems with large-scale datasets.

Figure 2.8. *A real-scenario dataset with pose variations. (a). Ground-Truth Face Images. (b)Face Template and Image to be aligned. Left Column: template image used to do alignment. The other columns are the Images to be aligned starting from the initial rectangle, labeled using green rectangle.*

Table 2.3. RMSPE results for the real-scenario dataset in Figure 2.8.

	Convergence results for images from dataset						
Person 1	0.01528	Diverged	Diverged	Diverged	Diverged	Diverged	Diverged
Person 2	0.01832	Diverged	Diverged	Diverged	Diverged	Diverged	Diverged
Person 3	0.01221	Diverged	Diverged	Diverged	Diverged	Diverged	Diverged
Person 4	0.1884	Diverged	Diverged	Diverged	Diverged	Diverged	Diverged

2.5.4 Complexity

The complexity of Directional Lucas-Kanade pyramid is evaluated on GPU GeForce 1080. Compared with traditional Lucas-Kanade pyramid method, in case of feature extraction in 4 directions, the computational cost is only 1.5 times of traditional LKP method (12ms vs. 8ms).

2.6 Conclusions

To alleviate the negative effect of noises in Face Alignment and Tracking, this chapter developed directional features of images and presents how directional features improve the accuracy of LK method in facial image alignment. Through the extraction of complete information in the image, including directional and scale information, the tracking accuracy can be significantly improved. The method provides additional robustness to transformations, translations as well as noises, yet it cannot work well when content is somehow lost or changed in case of large pose variations. Besides, through the parallel optimization on GPU, the directional method does not increase much computational complexity. The main material of this chapter mainly comes from Publication I.

3. Exploring Directional and Local Features for Face Recognition

3.1 Introduction

The face images are usually affected by different expressions, poses, occlusions and illumination changes, and the difference of face images from the same person could be larger than those from different ones. Thus, constructing an effective face representation is a key issue for face recognition.

Before deep methods emerge, many face representation approaches have been introduced, including subspace based holistic features and local appearance features. Holistic methods aim at reducing the high dimensionality of the raw face image space. They consider a face image as an entity and extracts only global feature from it. The shortcoming of the holistic methods is that they are highly sensitive to illumination and pose changes. Thus, holistic methods are always combined with local appearance features. In Chapter 2, directionality has been proved to be effective in face alignment. In this chapter, we further study the effectiveness of directionality in face recognition under noisy conditions.

In our work [142], directional information is assumed as one kind of global features, and the variant of LBP operator (mLBP) is applied to extract local information. First the directional information is extracted from the face image using directional filters. Directional filters are generated at global level using LPA technique. In order to extract the information at the local level mLBP operator is applied on directionally filtered face images after partitioning them into non-overlapping rectangular blocks of different sizes. Since the dimensionality of the obtained feature vector is quite high, Fisher Discriminant Analysis is used to reduce it. Finally, Support Vector Machine Classifier is used to perform the classification task.

The rest of this chapter is arranged as follows: In Section 3.2 we give a brief review of related works. In Section 3.3, we explore face representation based on directional and texture information. Section 3.4 gives the analysis of parameters when exploring directional features. Section 3.5 presents the experimental results on noisy datasets. We then conclude in Section 3.6.

3.2 Related Works

The hybrid approaches use both local and global features of the face image. In [72], Pentland *et al.* extended the idea of Eigenfaces to specific Eigenfeatures called Eigenmouth, Eigeneyes, etc. The combined representation of Eigenface and Eigenfeatures achieved better performance than the Eigenface based approach. In [73], Blanz *et al.* proposed a hybrid approach in which first the face image is decomposed into the different sections corresponding to facial features such as mouth and eyes. Further, a 3D model of face is used to generate face images under different illumination conditions and poses to train the classifier. In a number of approaches, part based face recognition techniques is applied after processing the face image in a holistic manner. For example, Zhang *et al.* [53] applied a set of Gabor filters on face images followed by block processing using LBP. However, the inter-scale dependences between various Gabor filtered images is not considered by this approach. Lei *et al.* [54] proposed Gabor Volume based LBP (GV-LBP) to address this problem. GV-LBP captures inter-scale dependences between adjacent scales and orientations of Gabor face images. These approaches [53][54][55][56][57][58] generally utilize a set of filters leading to forty Gaborfaces. Block processing of these Gaborfaces results in a very high dimensional face descriptor. Thus, Gabor filters based approaches are computationally expensive even when the image size is relatively small. In [74], Li *et al.* proposed to use heat kernels in order to extract the structural information from the face image followed by a texture extraction using LBP. This method achieved better results than those using the combination of Gabor and LBP, however, at the expense of complex mathematical operations on large matrices.

3.3 Face Representation based on Directionality and Texture

To classify a given face image, the algorithm contains the following steps: a) generating Face Pyramid (FP) from the original face image, see Section 3.3.1; b) capturing directional derivatives of the FP using single-scale LPA directional filters, constructing the Directional Derivative Pyramid (DDP), see Section 3.3.2; c) applying mLBP operator on the DDP to generate Directional mLBP Map Pyramid (DMMP), see Section 3.3.3; d) multiple-level block partitioning and feature concatenation, see Section 3.3.4; e) reducing the dimensionality of face features using LDA; f) training and classifying in face feature space using SVM classifier.

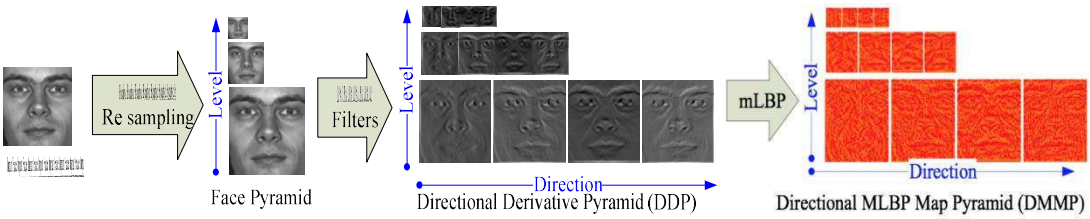


Figure 3.1. *Generating Directional Derivative Pyramid (DDP) and Directional MLBP Map Pyramid (DMMP) in case of four directions with fixed scale 3.*

3.3.1 Generating Face Pyramid (PF)

The pyramidal representation of a face image I is defined as a collection of resampled faces. The face image is resampled into multiple-resolution faces using Bicubic interpolation method [133].

Given a face image I of size $w \times h$, let $l = 1, 2, \dots, L$ be a generic pyramidal level, and the pyramidal representation is defined as follows. The 1st level image $I^1 = I$ is the highest resolution image (raw image) with sizes $w^1 = w$, $h^1 = h$. Denote I^l as the face image at level l, the width w^l and height h^l of l^{th} level face I^l are defined as the largest integers that satisfy the conditions:

$$w^l \le w/2^l, \quad h^l \le h/2^l \tag{3.1}$$

Then the l^{th} level face image I^l is obtained by down-sampling the raw image I from size (w, h) to size (w^l, h^l), and the Face Pyramid can be labelled as $\{I^l\}_{l=1}^{L}$..

3.3.2 Extracting Directional Derivative Pyramid (DDP)

Define h as the single scale value, N as the number of directions, then $\{\theta_i\}_{i=1}^{N}$ labels the collection of directions used in LPA filters. For direction θ_i with scale h, the first order derivative kernel $g_{h,\theta_i}^{(1)}$ is given as follows:

$$g_{h,\theta_i}^{(1)}(\boldsymbol{x}, \boldsymbol{X}_s) = (-1/h) w_{h,\theta_i}(\boldsymbol{x} - \boldsymbol{X}_s) \phi_h^T(\boldsymbol{x} - \boldsymbol{X}_s) \Phi_h^{-1} \phi^{(1)}(0) \tag{3.2}$$

$$\Phi_h = \sum_s w_{h,\theta_i}(\boldsymbol{x} - \boldsymbol{X}_s) \phi_h(\boldsymbol{x} - \boldsymbol{X}_s) \phi_h^T(\boldsymbol{x} - \boldsymbol{X}_s) \tag{3.3}$$

$$\phi_h(\boldsymbol{x}) = (-1)^{|\boldsymbol{k}|} \boldsymbol{x}^{\boldsymbol{k}} / \boldsymbol{k}!, \ \boldsymbol{k} \in Z^2 : \forall |\boldsymbol{k}| \le m \tag{3.4}$$

where $w_{h,\theta_i}(x - X_s)$ is the directional window, m is the order of polynomial, $\phi_h \in R^M$ is the complete set of linearly independent 2D polynomials of the powers from 0 till m, and the length M of the vector is:

$$M = (m+2)!/2 \cdot m! = (m+2)(m+1)/2 \tag{3.5}$$

Since directions $\{\theta_i\}_{i=1}^N$ define the sector division of local polynomial kernels while scale h defines the size of the support, the choice of directions and scale is crucial in LPA kernel design. Basically, directions should be chosen such that the selected $\{\theta_i\}_{i=1}^N$ are aligned with the majority of the prominent features. And scale h should define the most reasonable support used to calculate the directional derivatives. A detailed analysis of selecting these parameters is given in Section 3.4.

For l^{th} level face $I^l(\boldsymbol{x})$, the directional derivative $D^l_{h,\theta_i}(\boldsymbol{x})$ for direction θ_i and scale h is given by:

$$D^l_{h,\theta_i}(\boldsymbol{x}) = I^l(\boldsymbol{x}) \circledast g^{(1)}_{h,\theta_i}(\boldsymbol{x}) \tag{3.6}$$

where $\circledast$ indicates convolution operator.

For each level face in the pyramid, totally N directional derivatives are obtained using equation (3.6), labeled by $\{D^l_{\theta_i}\}_{i=1}^N$. Then the directional derivative pyramid $\{D^l_{\theta_i}\}_{i=1,\dots,N;l=1,\dots,L}$ can be simply obtained by applying equation (3.6) to all faces $\{I^l\}_{l_p=1}^L$ in the pyramid. Taking four directions $\theta_i \in \{0, \pi/4, \pi/2, 3\pi/4\}$ and pyramid level $L = 3$ for instance, DDP of a face image is presented in figure 3.1.

3.3.3 Generating Directional mLBP Map Pyramid (DMMP)

After obtaining DDP, texture-based information would be further extracted to represent the derivative pyramid. In this thesis, an alternative form of LBP, modified LBP (mLBP) operator is applied on a 3x3 neighborhood. Let g_c label the central pixel, and $\{g_0, g_1, \dots, g_7\}$ are the 3x3 neighborhood around g_c. Then the mLBP operator is defined by:

$$mLBP(g_c) = \sum_{j=0}^{3} s(g_j - g_{j+4}) \cdot 2^j + s(g_c - g_M) \cdot 2^4 \tag{3.7}$$

where $g_M = (g_0 + g_1 + \dots + g_7)/8$ is the mean value of the neighborhood, and $s(x)$ is the step function:

$$s(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{3.8}$$

This mLBP descriptor reduces the number of bins to 32 which is much less than 256 in original LBP. It encodes the pixels of an image by thresholding the 3x3-neighborhood of each pixel with the center value and considering the result as a binary number. By applying mLBP operator on the derivatives in DDP, the directional mLBP maps are obtained for all faces in the pyramid: $DMMP = \{M_{\theta_i}^l\}_{i=1,\ldots,N;\ l=1,\ldots,L}$, see figure 3.1.

3.3.4 Block Partitioning and Feature Concatenation

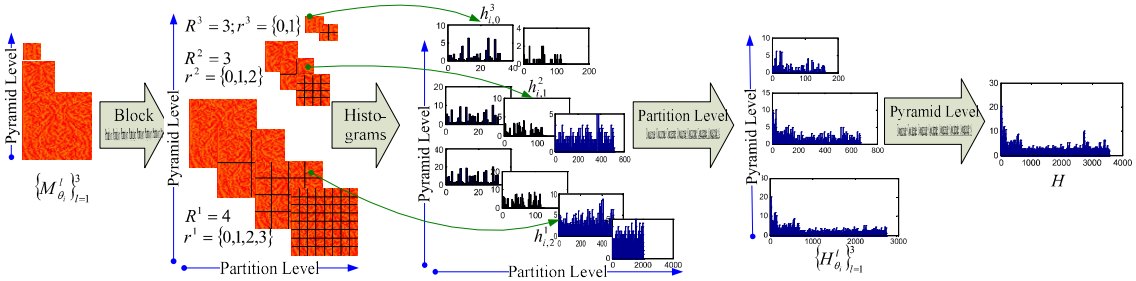


Figure 3.2. *Pyramidal Block Partitioning and Feature Concatenation for DDPs.*

To construct descriptors representing input faces in feature space, the maps are further partitioned into multi-resolution blocks and the block-wise histograms are computed then concatenated together. Taking direction θ_i for instance, the textural features are extracted from each map in the pyramid separately and then concatenated together, as shown in figure 3.2.

For each level derivative image $D_{\theta_i}^l$ in DDP, its corresponding mLBP map will be $M_{\theta_i}^l$. We then build multi-resolution frameworks by placing a sequence of increasingly coarser grids over the map and then giving a weighted sum of the number of matches computed from each level of resolution.

Specifically, for l^{th} level mLBP map $M_{\theta_i}^l$ in DMMP, the map is spatially partitioned into blocks at various levels. Here we define $r^l = \{0, 1, \ldots, R^l - 1\}$ to represent the Multi-Resolution Partitioning Levels (MRPLs) when dividing the l^{th} level mLBP map into multi-resolution blocks, where R^l is the maximum partitioning level. Then the block partitioning is defined as follows. The lowest partitioning level $(r^l = 0)$ takes the whole image as a single block, i.e. no partitioning. As r^l increases from 0 to $R^l - 1$, the partitioning is done by further dividing the lower level blocks into 4 blocks.

To summarize, for any level $r^l > 0$, the image is partitioned into 4^{r^l} non-overlapping blocks.

From pyramidal view, the MRPLs r^l is determined by Maximum Partitioning Levels (MPLs): $R^l = R^1, R^2, \dots, R^L$. Specifically, the value of R^l decreases as l increases, which means that the partitioning gets coarser as the map shrinks. Taking pyramid level $L = 3$ for instance, R^l and r^l are defined as following:

$$R^1 = 4, R^2 = 3,\ R^3 = 2 \tag{3.9}$$

$$r^1 = \{0,1,2,3\}, r^2 = \{0,1,2\},\ r^3 = \{0,1\} \tag{3.10}$$

Histograms obtained from blocks at partitioning level r^l are concatenated together and represented as $h_{i,r}^l$. Given that coarser levels contain more dissimilar information; a weight vector is introduced to penalize the number of features at the coarser levels:

$$w_r^l = 1/2^{R^l - r^l} \tag{3.11}$$

For a Directional mLBP Map (DMM) $M_{\theta_i}^l$ in the pyramid, the features of different partition levels $h_{i,r}^l$ are concatenated together:

$$H_{\theta_i}^l = \left\{w_0^l \cdot h_{i,0}^l, w_1^l \cdot h_{i,1}^l, \dots, w_{R-1}^l \cdot h_{i,R-1}^l\right\} \tag{3.12}$$

Same procedure is done on all N DMMs and the l^{th} level feature vectors are concatenated as follows:

$$H^l = \left\{H_{\theta_1}^l, H_{\theta_2}^l, \dots, H_{\theta_N}^l\right\} \tag{3.13}$$

Final descriptor of the face I will be the concatenation of all L--Level vectors:

$$H = \{H^1, H^2, \dots, H^L\} \tag{3.14}$$

The dimensionality of the face descriptor is given by:

$$Dimensionality = \sum_{l=1}^{L}\left(N \times 32 \times \sum_{r^l=0}^{R^l-1} 4^l\right) \tag{3.15}$$

where L is the pyramid level, N is number of directions, 32 is the number of bins in mLBP descriptor, and R^l is the maximum partitioning level when dividing the l^{th} level map into multi-resolution blocks.

As can be seen from equation (3.15), the proposed pyramidal feature extraction method leads to a high dimensionality. In order to reduce the dimensionality, the LDA based technique [60] is applied. Then face recognition is performed using a Support Vector

Machine (SVM) classifier [134]. A multiclass SVM classifier with linear kernels is learned by using the libSVM tool [75]. As a result, several support vectors are identified and stored for the use of prediction. Then the test samples are matched with these identified support vectors to label the predicted class.

3.4 Constructing LPA Kernels

In this section, we mainly focus on how kernel construction influences the recognition accuracy of the proposed method. The effect of extracting multi-level pyramidal information will be discussed in Section 3.5. The construction of LPA kernels $g_{h,\theta_i}^{(1)}$ mainly depends on scale size h and the division of support $\{\theta_i\}_{i=1}^{N}$. As a result, the performance of the algorithm is crucially influenced by these parameters.

To study the effect of these parameters, non-symmetric uniform directional window $w_{h,\theta}$ is selected, and the polynomial order m used for generating LPA kernels equation (3.2-3.5) is set to $m = [1, 0]$, meaning that totally 3 polynomials $\phi_h = [\phi_0, \phi_1, \phi_2]$ are utilized, with $\phi_0 = 0$, $\phi_1 = x_1$, $\phi_2 = x_2$. Besides, a directional information of only original-size image is considered by setting the pyramid level $L = 1$. The multi-resolution block partitioning level is $r^1 = \{0, 1,2,3\}$, i.e. the maximum partitioning level $R^1 = 4$. Experiments are performed on face images from the Yale [9] and Extended YaleB [135] dataset.

The Yale Face Database [9], published by the Yale University in 1997, contains 160 frontal face images in total. These images are from 16 people with 10 kinds of settings: an image under ambient lighting, one with or without glasses, three images under different light sources, and five images with different expressions. These images are all gray-scale with resolution of 320x243.

The Yale Face Database B [135], also published by the Yale University, but in 2001, extended the original Yale Face dataset by including face images with large variation of pose and illumination. Extended Yale-B dataset [136] consists of 16128 images of 28 individuals and YaleB dataset consists of 5760 face images of ten individuals. These face images are all gray-scale as the images in the Yale Database, but with a resolution of 640 x 480.The images are taken under 9 different pose and 64 different illumination conditions. These two datasets can be used to verify the robustness of each method against pose variations and different illuminations.

3.4.1 Choice of Directions

To generate the directional kernel for a particular direction θ_i, we need to rotate the kernel support in order to direct it to the desired direction. Derivatives in different directions reflect diverse textural information of a face image. Recognition tests are performed for several cases: dividing the support into 2 directions, 4 directions and 8 directions within ranges $[0,\pi]$ and $[0,2\pi]$. These divisions are labeled by $D_{\pi,N} = \{i \cdot \pi/N\}_{i=1}^{N}$ and $D_{2\pi,N} \in \{i \cdot 2\pi/N\}_{i=1}^{N}$ respectively. The accuracy and complexity performance are shown in Table 3.1. The time consumption is shown for the classification of 2224 test face images of YaleB [135] dataset which are trained using 5 training samples for each individual.

Table 3.1. *Influence of Support Division.*

Directions	$D_{\pi,2}$	$D_{\pi,4}$	$D_{\pi,8}$	$D_{2\pi,2}$	$D_{2\pi,4}$	$D_{2\pi,8}$
Accuracy (%)	89.26	93.14	94.15	82.89	90.01	93.53
Dimensionality	5440	10880	21760	10880	5440	21760
Time (sec)	44.07	86.49	174.69	44.25	86.57	174.53

Generally speaking, the more diverse directions are utilized when extracting directional derivatives, the more directional information is covered in the final face descriptor H, see equation (3.13-3.14). As a result, the descriptor H will have a larger dimensionality and higher classification accuracy. Thus a compromise shall be made between the accuracy and complexity. Directions should be chosen such that the selected $\{\theta_i\}_{i=1}^{N}$ are aligned with the majority of the prominent features without redundancy. As shown in Table 3.1, finer division of the support provides better accuracy results. Due to the redundancy in directions introduced when partitioning the support within $[0, 2\pi]$, better performance is provided by selecting directions between 0 and π, i.e. $D_{\pi,N}$ is superior over $D_{2\pi,N}$. A compromise is made between accuracy and complexity by selecting $D_{\pi,4}$ as the support division.

Table 3.2. *Influence of Scales*

Scale	2	3	4	5	6	7	8
Accuracy (%)	91.98	93.14	91.42	89.85	89.44	86.42	83.22

3.4.2 Choice of Scale

Scale h defines the size of the window, which is also the number of neighboring nodes used to filter the image. Too large or too small h will result in over-smoothing or under-smoothing derivatives respectively. Thus the optimum scale should be carefully selected. Table 3.2 shows that smaller scales provide better results, and best performance is given when scale h equals 3. Besides, the accuracy degrades as the image gets more and more over-smoothed. This can be partly attributed to the fact that the dataset used for evaluation is noise-free. A larger scale shall be selected if noise is present in the dataset.

3.5 Experimental Results

To evaluate the performance of the proposed algorithm, tests are conducted using a number of publicly available face datasets, including YaleB [135], Extended YaleB [136], ORL (http://www.cam-orl.co.uk), CMU-PIE [137], AR [13], and FERET [138][139]. Comparison is made with respect to several state-of-the-art methods.

ORL dataset consists of images taken from 40 different individuals with 10 images of each person. The images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not-smiling) and facial details (glasses/no-glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with a tolerance for some side movement).

CMU PIE dataset consists of 41,368 images of 68 people. Each person is captured under 13 different poses, 43 illumination conditions, and 4 expressions. The dataset has extreme illumination variations and a large number of images for each subject.

AR dataset consists of more than 4000 images of 126 individuals of which 70 are male and 56 are female. In AR dataset the images were taken in two sessions separated by two weeks, considering expression (neutral, smile, anger and scream) and occlusion (sunglass and scarf) variations. In our experiments we take images of 45 men and 45 women.

FERET dataset contains frontal face images divided into five categories: Fa, Fb, Fc, Dup1, and Dup2. Fb images were taken at the same day as Fa images and with the same camera and illumination condition. Fc images were taken at the same day as Fa pictures but with different cameras and illumination. Dup1 images were taken on different days

than Fa but within a year. Dup2 images were taken at least one year later than Fa. In the FERET tests, 1196 Fa face images are gallery samples for which the labels are known. 1195 Fb, 194 Fc, 722 Dup1, and 234 Dup2 pictures are probe set for which the labels are not known. The number of images in Fa set indicate the number of classes, so for each person there is just one training image.

Each of these datasets simulates a scenario which causes variation in the face image in real time face recognition application. Using these datasets, we evaluate the performance of proposed method in different test scenarios such as, expression variation, illumination variation, occlusion, Gaussian noise corrupted face images and face images with missing pixels.

3.5.1 YaleB and Extended YaleB Datasets

Our experiment images with frontal pose of YaleB [135] and Extended YaleB [136] with all 64 different illumination settings are utilized. Different numbers of training samples are utilized in the evaluation process, and comparison is made with respect to PCA, LDA, LBP [140], CDA [76] and CEA [77].

To illustrate the effect of pyramidal construction, pyramid levels $L = 1,2,3$ with MRPLs $r^1 = \{0,1,2,3\}, r^2 = \{0,1,2\}$, $r^3 = \{0,1\}$ are evaluated. The classification accuracies are shown in Table 3.3. It can be observed that the proposed algorithm outperforms all other approaches. Special attention should be paid to the fact that pyramidal construction provides additional progress in case of insufficient trainings.

Table 3.3. *YaleB and Extended YaleB Datasets*

Num. of Train	**Methods**					**Ours**		
	PCA	LDA	LBP [140]	CDA [76]	CEA [77]	L=1	L=2	L=3
5	45.27	63.87	50.35	73.06	77.39	93.14	93.06	<u>94.16</u>
10	63.94	80.60	73.93	89.81	92.45	98.90	98.93	<u>98.95</u>
20	68.78	73.04	89.69	88.34	93.78	<u>99.62</u>	99.60	99.60
30	72.29	78.33	95.21	87.46	96.44	<u>99.71</u>	99.69	99.69

(The results to be compared with are taken directly from [77])

3.5.2 CMU-PIE Dataset

Best performance is achieved when we first resample the faces to 64×64 and then construct a 3-level pyramid using MRPLs. The accuracy results are shown in Table 3.4. Our method achieves an accuracy of almost 90% for 5 training samples, and the accuracy is even over 99% for 20 training samples. The huge progress can also be attributed to the fact that the face descriptor generated using directional derivative pyramids is robust to illumination variations.

Table 3.4. *CMU-PIE Datasets*

Num. of Train	**Methods**					**Ours**		
	PCA	LDA	LBP [140]	CDA [76]	CEA [77]	L=1	L=2	L=3
5	34.34	46.65	55.87	59.03	64.71	89.19	89.83	<u>89.96</u>
10	36.46	69.67	77.32	78.72	81.13	96.66	97.16	<u>97.25</u>
20	52.84	84.24	90.17	89.6	90.55	98.47	98.96	<u>99.08</u>

(The results to be compared with are taken directly from [77])

3.5.3 AR Dataset

Table 3.5. *AR Datasets*

Methods		**Variations**		
		expression	Sunglasses	scarf
PCA		74.07	12.96	2.41
LDA		72.41	11.85	9.81
LBP		87.04	34.63	47.04
LGBP-M [53]		86.11	37.59	82.59
GV-LBP-TOP-M [54]		90.56	53.89	<u>87.41</u>
E-GV-LBP-M [57]		90.93	42.77	82.78
E-GV-LBP-P [57]		89.81	44.07	86.67
Ours	L=1	91.85	95.37	77.96
	L=2	92.78	<u>96.30</u>	79.26
	L=3	<u>93.52</u>	<u>96.11</u>	81.30

(The results to be compared with are taken directly from [57])

Table 3.5 compares the recognition rates of different methods on AR expression, sunglass and scarf occlusion sets, respectively. Best performance is achieved when we

first resample the face to 40x40, and then construct a 3-level pyramid. It can be seen that for expression and sunglasses tests, the proposed method has highest recognition rates. Compared to other methods, the accuracy almost doubles for sunglasses test. For scarf test, although our algorithm does not provide the highest recognition accuracy, the contribution of constructing directional derivative pyramids is obvious. The reason is that scarf disturbs the overall structure of the face image and extracting smaller-size faces when building up face pyramids alleviates such influences.

3.5.4 FERET Dataset

Table 3.6. *FERET Datasets*

Methods		Fb	Fc	Dup1	Dup2
PCA		78.91	9.79	33.66	11.97
LDA		87.78	47.42	44.32	20.09
LBP		97.00	79.00	66.00	64.00
LGBP-M [53]		98	97	74	71
GV-LBP-TOP-M [54]		98.08	98.45	80.89	81.20
E-GV-LBP-M [57]		98.41	98.97	81.99	81.62
E-GV-LBP-P [57]		97.82	97.42	81.99	78.63
Ours	L=1	97.65	96.91	83.24	76.50
	L=2	97.74	97.94	84.76	88.03

(The results to be compared with are taken directly from [57])

Best performance is provided when we resize the faces to 60×60 and reset the number of blocks for each partitioning levels as following: The number of blocks is set to 1×1, 3×3, 5×5 and 10×10 for levels $r^l = 0, 1, 2, \text{and } 3$ respectively. The MRPLs are set to $r^1 = \{0,1,2,3\}, r^2 = \{0,1,2\}$, $r^3 = \{0,1\}$. Comparison is made with several methods as shown in Table 3.6.

3.5.5 Noisy ORL

In this section we evaluate the performance of the proposed method on the noisy face image. Considering two different scenarios: a) the additive Gaussian noise is assumed to be present in the face images b) some pixels are missing from the face image. The second case can be considered also as a case of having negative impulse noise.

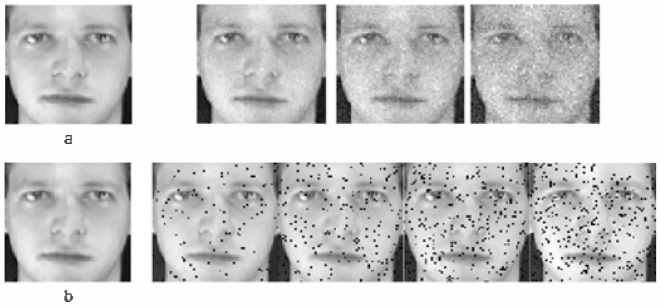
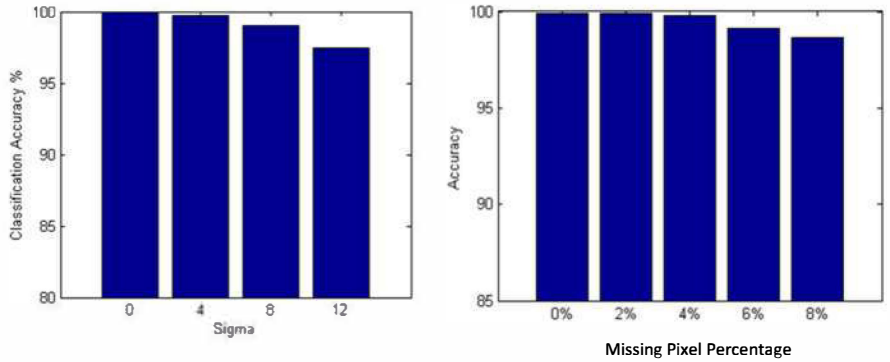


Figure 3.3. *Face Images with Noises. (a) A face image and its corresponding Gaussian noisy images with* $\sigma = 4$*, 8 and 12. (b) A face image and next to it are noisy images with some missing pixels. The percentage of missing pixels is 2, 4, 6 and 8 %.*



(a) Gaussian Noise (b) Impulse Noise

Figure 3.4. *Accuracy Performance in case of Noises. (a)The accuracy of ORL dataset when different amount of Gaussian noise is added to it. (b) Accuracy for ORL dataset when a certain percent of pixels is missing from the image.*

Gaussian noise: Varying amount of Gaussian noise is added to ORL dataset with the standard deviations σ = 4, 8, and 12. The noisy face image examples with different sigma are shown in figure 3.3. Figure 3.4 (a) shows how the classification accuracy varies on ORL dataset as σ increases from 0 to 12. It can be observed that even with sufficient amount of noise, the classification accuracy does not change abruptly.

Impulse Noise: Case of missing pixels is considered by adding negative impulse noise to face images. The amount of noise is varied by measuring the percent of missing pixels which are varied here from 0 to 8%. The image examples with missing pixels are shown in figure 3.3. The classification accuracy of the proposed method with respect to the percentage of missing pixels is shown in figure 3.4(b) The accuracies for the different corrupted images were 99.925, 99.90, 99.775, 99.15, and 98.65, respectively. It can be seen that till 6% of missing pixels the accuracy remains above 99% and the drop in the accuracy with increasing number of missing pixels is then quite gradual.

3.5.6 Complexity Analysis

Table 3.7. *Accuracy and Computational Complexity as partitioning level increases.*

Partitioning Levels	1	2	3	4
Accuracy (%)	8.94	35.90	77.13	93.14
Dimensionality	128	640	2688	10880
Time (sec)	13.11	22.27	32.81	86.49

Table 3.8. *Accuracy and Computational Complexity as pyramid level increase.*

Pyramid Levels	1	2	3
Accuracy (%)	93.14	93.06	94.16
Dimensionality	10880	13568	14208
Time (sec)	86.49	103.12	112.31

The level of partitioning is an important factor which maintains a balance between the dimensionality and the extraction of part based features. With higher partitioning level, there is an increase in the number of blocks resulting in higher dimensionality. Computational complexity also increases with the partitioning level. Table 3.7 shows the classification performance as the levels increase with respect to the time. The time is shown for the classification of 2224 test face images of YaleB dataset which are trained using 5 training samples for each individual. The lower levels encode the holistic information of the face while the higher level encodes the part based information. The higher levels vary more across the different face images and consist of more discriminative information.

Other than partitioning levels, pyramid level also has significant influence on the accuracy and complexity. With higher pyramid level, there is a non-significant increase in the number of blocks resulting in higher dimensionality. Computational complexity also increases with the pyramid level. Table 3.8 shows the classification performance as the levels increase with respect to the time.

3.6 Conclusions

To tackle the problem of appearance noise in face images, including Gaussian white noise and missing pixels, this chapter presents a novel approach to face recognition

problems utilizing directional and textural information of the faces in a pyramidal manner. The face image is first resampled into multiple-resolutions to construct a face pyramid. Then single-scale directional LPA filters are utilized on the face pyramid to extract the directional derivative pyramids. Multi-Resolution mLBP operator is then applied on the derivatives to capture both local and global features. The pyramidal structure of the proposed method provides further robustness especially when the face image has structure disturbance. Besides, through the utilization of LPA filters, the face recognition method gains robustness with respect to noises. The main material of this chapter comes from Publication II.

4. Face Clustering for Unlabeled Datasets

4.1 Introduction

Multi-view learning becomes increasingly attractive due to the explosion of multi-view data, including samples represented by different feature descriptors, and objects from multiple sources, e.g., text, image, audio and video [33]. To exploit the complementary information from multiple views, numbers of multi-view learning methods have been developed and achieve superior performances in comparison with single-view learning. Analogous to other machine learning tasks, multi-view learning also suffers from two phenomena in the big data era, i.e., the lack of label information and high-dimensionality of the feature space [33]. The high expense of labelling data and the data explosion make most data unlabeled. Without label information, multi-view learning becomes more challenging in the unsupervised case [34].

Generally, it is intuitive to concatenate the multi-view data directly, and thus the traditional single-view solutions can be applied. Whereas, a simple concatenation will aggravate the curse of dimensionality, ignore the complementary nature and destroy distinct statistical properties of different views [35]. For unsupervised learning, one key challenge is how to discover the data structure by clustering. Multi-view clustering extends the clustering techniques in single-view learning (e.g. spectral clustering [36][37][38][39], linear regression [40] and matrix factorization [41]) to multi-view tasks. The common part of different views is usually modelled by cluster indicator matrix and diverse graph learning algorithms are developed. To alleviate the effect arising from the curse of dimensionality, dimension reduction algorithms are proposed to project the samples to a low-dimensional feature subspace. Thus, the time complexity and the computation burden are reduced while the generalization ability of the learning machines can be improved [42]. Traditional methods, like CCA [43], [44] and PLS [45], can be used to cope with the two-view case. To exploit the correlations of multiple views simultaneously, researchers proposed many multi-view dimension reduction techniques [34][35][46][47][48]. Multi-view dimension reduction shares many

similarities with multi-view clustering, and adopts similar techniques (e.g., matrix factorization and spectral analysis) to learn a projection matrix for each view.

For multi-view unsupervised learning, the key issue is to learn a latent representation for all views. In unsupervised learning, sample similarity relationships are often used in learning the latent space for both clustering and dimension reduction. For matrix factorization based methods, the latent space should well recover the raw feature space. However, almost all existing models ignore the predictive ability of the feature space, which is quite important for unsupervised learning. An undirected latent space Markov network was proposed to discover a predictive latent subspace representation shared by multiple views [49]. A latent space learning model was proposed to connect the feature space and label space for multi-label classification with many classes [50]. Nevertheless, they are specially designed for supervised learning.

In this chapter, we propose a novel multi-view predictive latent space learning model (MVP) for multi-view unsupervised learning [143]. MVP learns a latent predictive representation by maximizing the correlation between the feature space of each view and the latent space. As the latent space is shared by all views, the consensus principle is fully used for multi-view learning. Considering the complementary nature, MVP learns a weighted graph to preserve the locality in multiple feature spaces jointly. Experimental results on datasets with multiple features show that MVP is superior to the state-of-the-art multi-view clustering algorithms.

4.2 Related Works

In this section, we will briefly review two typical learning tasks in multi-view unsupervised learning.

Multi-view clustering algorithms have been developed to cluster data from multiple views simultaneously, by deriving a solution which uncovers the common latent structure shared by multiple views. Spectral clustering makes use of the spectrum of the similarity matrix of the data to discover the hidden data clusters [51]. Co-regularized

multi-view spectral clustering (Co-regSC) is a multi-view spectral clustering framework by co-regularizing the clustering hypotheses [36]. Multi-modal spectral clustering (MMSC) considers each type of feature as a modal, and integrates such heterogeneous features by learning a commonly shared graph Laplacian matrix [37]. Multi-view non-negative matrix factorization (MultiNMF) is a NMF-based multi-view clustering algorithm, and it formulates multi-view learning as a joint matrix factorization process [41]. [40] integrated all features of different views and used joint structured sparsity-inducing norms to learn a weight for each feature. Multi-view spectral clustering (MVSC) is a large-scale approach based on the bipartite graph to solve the massive data problem [38]. Auto-weighted multiple graph learning (AMGL) is a parameter-free multi-view model that can learn an optimal weight for each view automatically [39].

In multi-view unsupervised learning, due to the lack of label information, the latent representation shared by multiple views, is usually expected to be discriminative and predictive. Whereas, the current matrix factorization and spectral analysis based methods do not emphasize the predictability of the latent space. In this chapter, we will investigate the learning of a predictive latent space for multi-view unsupervised tasks.

4.3 Multi-View Predictive (MVP) Latent Space Learning

4.3.1 Model

Let $\mathbf{X} = [\mathbf{X}_1, ..., \mathbf{X}_i, ..., \mathbf{X}_K]$ be a set of multi-view data. $\mathbf{X}_i$ is the data matrix of the i^{th} view. As shown in figure 4.1, multi-view unsupervised learning aims to learn a latent representation for all views. There are two important principles in multi-view learning, i.e., the consensus principle and the complementary principle [33]. Whereas, because there is no class information, unsupervised learning in multi-view tasks becomes much challenging.

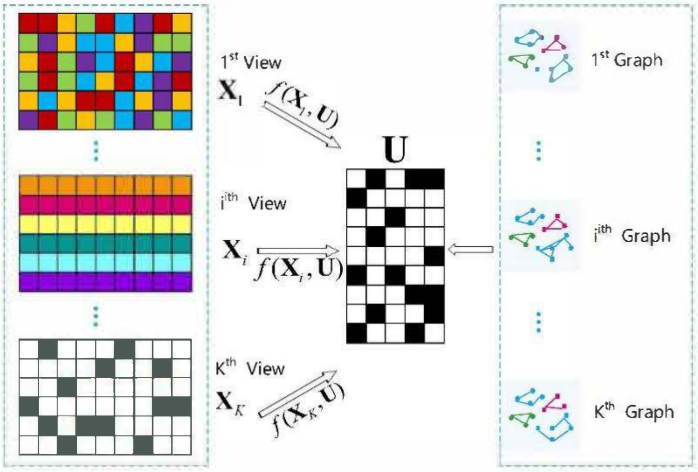


Figure 4.1. *The framework of multi-view predictive latent space learning (MVP).*

Therefore, a latent representation with outstanding predictability is expected. Assume that $\mathbf{U} \in \mathbb{R}^{n \times h}$ is the latent space for all views, where h is the dimension of the latent space. The latent space is a low-dimensional representation of each view that can well characterize the discriminant structure embedded in multi-view high-dimensional data. Inspired by the definition of the predictability for the latent space in [50], we formulate the predictability of $\mathbf{U}$ by the correlation between the i^{th} feature space and the latent space as

$$f(\mathbf{X}_i, \mathbf{U}) \tag{4.1}$$

where f is a function to measure the correlation between $\mathbf{X}_i$ and $\mathbf{U}$. The correlation $f(\mathbf{X}_i, \mathbf{U})$ should be maximized to enhance the predictability of the latent space $\mathbf{U}$. We use $\boldsymbol{u}$ to represent a column of $\mathbf{U}$, Therefore, the correlation between the feature space $\mathbf{X}_i$ and $\mathbf{u}$ can be represented by $f(\mathbf{X}_i, \boldsymbol{u})$.

$$f(\mathbf{X}_i, \mathbf{U}) = \frac{(\mathbf{X}_i \boldsymbol{v})^T \boldsymbol{u}}{\sqrt{(\mathbf{X}_i \boldsymbol{v})^T \mathbf{X}_i \boldsymbol{v}} \sqrt{\boldsymbol{u}^T \boldsymbol{u}}} \tag{4.2}$$

Where $\boldsymbol{v}$ is a linear projection for $\mathbf{X}_i$. The orthogonal constraint is imposed on the latent space $\mathbf{U}$, i.e.,$\mathbf{U}^T\mathbf{U} = \mathbf{I}$. Then the following optimization problem is introduced to maximize $f(\mathbf{X}_i, \boldsymbol{u})$.

$$\max_{\boldsymbol{u}} (\mathbf{X}_i \boldsymbol{v})^T \boldsymbol{u} \;\; s.t. (\mathbf{X}_i \boldsymbol{v})^T \mathbf{X}_i \boldsymbol{v} = 1 \tag{4.3}$$

When $\boldsymbol{u}$ is fixed, we use the method of Lagrange multipliers to calculate the optimal

$\boldsymbol{v}$, denoted as $\widehat{\boldsymbol{v}}$:

$$\widehat{\boldsymbol{v}} = \frac{(\mathbf{X}_i{}^T\mathbf{X}_i)^{-1}\mathbf{X}_i{}^T\boldsymbol{u}}{\sqrt{\boldsymbol{u}^T\mathbf{X}_i(\mathbf{X}_i{}^T\mathbf{X}_i)\mathbf{X}_i{}^{-1}\boldsymbol{u}}} \tag{4.4}$$

Then the maximal $(\mathbf{X}_i\boldsymbol{v})^T\boldsymbol{u}$ can be derived as:

$$(\mathbf{X}_i\widehat{\boldsymbol{v}})^T\boldsymbol{u} = \sqrt{\boldsymbol{u}^T\boldsymbol{\Lambda}_i\boldsymbol{u}} \tag{4.5}$$

where $\boldsymbol{\Lambda}_i = \mathbf{X}_i\left(\mathbf{X}_i{}^T\mathbf{X}_i\right)^{-1}\mathbf{X}_i{}^T$. Hence, to improve the predictability of the latent space, each column u of **U** is supposed to satisfy equation (4.5). The objective function for the total predictability of the latent space $\mathbf{U}$ in multi-view learning can be formulated as:

$$\max_{\mathbf{U}} \sum_{i=1}^{K} tr(\mathbf{U}^T\boldsymbol{\Lambda}_i\mathbf{U}) \;\; s.t.\, \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{4.6}$$

Equation (4.6) takes the consensus principle into account by learning a common representation for all views. However, it does not consider the local geometry structure of the feature space in each view and the complementary information is not exploited as well. Hence, we introduced a weighted multigraph regularization for the latent space. Then, we model manifold regularized latent space learning as:

$$\max_{\mathbf{U},\mathbf{w}} \sum_{i=1}^{K} tr(\mathbf{U}^T(\Delta_i^* + \alpha w_i^r\mathbf{L}_i^*)\mathbf{U})$$

$$s.t. \sum_{i=1}^{K} w_i = 1, w_i \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{4.7}$$

where $\mathbf{L}_i^* = \mathbf{D}_i^{-1/2}\mathbf{W}_i\mathbf{D}_i^{-1/2}$ is the Laplacian matrix for the i^{th} view, w_i is the weight for the i^{th} view and $\alpha > 0$ is a tradeoff parameter, which adjusts the portion and thus the influence of Δ_i^* and $w_i^T\mathbf{L}_i^*$. And we normalize $\boldsymbol{\Lambda}_i$ just like the Laplacian matrix $\mathbf{L}_i$ and denote it by Δ_i^*. In equation (4.7), we will later introduce a weight for each view, with a parameter $r > 0$. However, for the maximization problem, it is obvious that only the w_i of the maximum $tr(\mathbf{U}^T\mathbf{L}_i^*\mathbf{U})$ is close to 1, and others are 0. It means that only one view is selected by this method, which does not coincide with our objective on exploring the complementary property of multiple views. Hence, we transform the objective function into a minimization problem:

$$\min_{U,w} \sum_{i=1}^{K} tr(\mathbf{U}^T(\Delta_i + \alpha w_i^r \mathbf{L}_i)\mathbf{U})$$

$$s.t. \sum_{i=1}^{K} w_i = 1, w_i \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{4.8}$$

Where $\Delta_i = \mathbf{I} - \Delta_i^*$, $\mathbf{L}_i = \mathbf{I} - \mathbf{L}_i^*$, and $\mathbf{I}$ denotes an identity matrix.

In this chapter, we adopt a parameter r to modulate the effect of the smoothness difference of graphs. The same phenomenon will occur when $r = 1$: only w_i of the minimum $tr(\mathbf{U}^T\mathbf{L}_i\mathbf{U})$ is close to 1, and other entries in $\mathbf{w}$ is 0, thus only one view is selected. To avoid this problem, we set $r > 1$.

4.3.2 Optimization Solution and Analysis of Model Complexity

In this section, we summarize the detailed optimization procedures of the model. We use an alteration minimization method to solve the optimization problem in equation (4.8). Then we update $\mathbf{U}$ and $\mathbf{w}$, respectively.

Sub-problem $\mathbf{U}$ By fixing $\mathbf{w}$ and omitting the irrelevant items with respect to $\mathbf{U}$, the objective function can be written as:

$$\min_{\mathbf{U}} tr(\mathbf{U}^T \sum_{i=1}^{K} (\Delta_i + \alpha w_i^r L_i)\mathbf{U})$$

$$s.t. \mathbf{U}^T\mathbf{U} = \mathrm{I} \tag{4.9}$$

For the optimization problem in equation (4.9), we can decompose each column $\boldsymbol{u}$ of the matrix $\mathbf{U}$ into an optimization subproblem. Each sub-problem should satisfy the following condition when we introduce a Lagrange multiplier to solve the problem with an equality constraint.

$$\sum_{i=1}^{K} (\Delta_i + \alpha w_i^r \mathbf{L}_i)\boldsymbol{u} = \lambda \boldsymbol{u} \tag{4.10}$$

where λ is the introduced Lagrange multiplier for $\boldsymbol{u}$. Hence, we transform the optimization for $\mathbf{U}$ to a general eigenvalue problem.

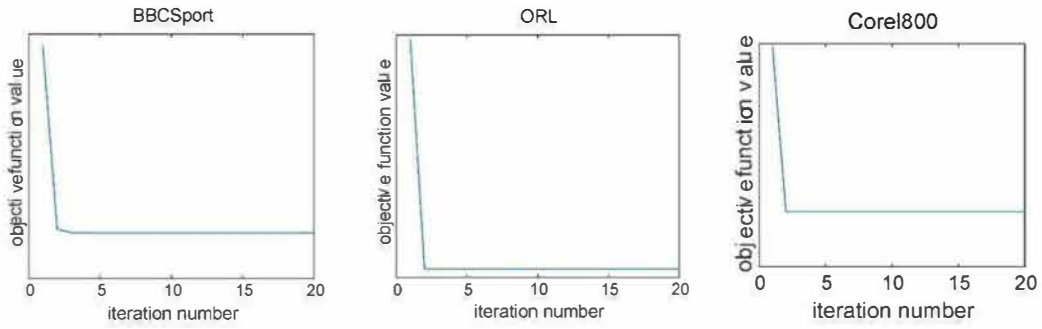


Figure 4.2. *The convergence curve of MVP for all the datasets.*

Sub-problem w When **U** is fixed, the optimization problem becomes only relevant to **w**. The objective function degenerates into:

$$\min_{\mathbf{w}} tr(\mathbf{U}^T \sum_{i=1}^{K}(\Delta_i + \alpha w_i^r \mathbf{L}_i)\mathbf{U})$$

$$s.t.\ \sum_{i=1}^{K} w_i = 1, w_i \geq 0 \tag{4.11}$$

By using a Lagrange multiplier ξ to take the constraint $\sum_{i=1}^{K} w_i = 1$ into consideration, we get the Lagrange function as follows:

$$\mathcal{L}(\mathbf{w}, \xi) = \begin{Bmatrix} tr(\mathbf{U}^T \sum_{i=1}^{K}(\Delta_i + \alpha w_i^r \mathbf{L}_i)\mathbf{U}) \\ -\xi(\sum_{i=1}^{K} w_i - 1) \end{Bmatrix} \tag{4.12}$$

By setting the derivative of $\mathcal{L}(\mathbf{w}, \xi)$w.r.t. w_i and ξ to zero, we have:

$$\begin{cases} \frac{\partial \mathcal{L}(\mathbf{w},\xi)}{\partial w_i} = r\alpha w_i^{r-1} tr(\mathbf{U}^T \mathbf{L}_i \mathbf{U}) - \xi = 0 \\ \frac{\partial \mathcal{L}(\mathbf{w},\xi)}{\partial \xi} = \sum_{i=1}^{K} w_i - 1 = 0 \end{cases} \tag{4.13}$$

After equation (4.13) is solved, the updating formula for w_i can be obtained:

$$w_i = \frac{\left(1/tr(\mathbf{U}^T \mathbf{L}_i \mathbf{U})\right)^{1/(r-1)}}{\sum_{i=1}^{K}\left(1/tr(\mathbf{U}^T \mathbf{L}_i \mathbf{U})\right)^{1/(r-1)}} \tag{4.14}$$

The Laplacian matrix $\mathbf{L}_i$ is positive semi-definite, so we have $w_i \geq 0$ naturally. According to equation (4.14) we can find that r modulates the effect of the smoothness difference of graphs.

Time complexity We use an alternation maximization strategy for the proposed MVP. The main computation burden lies in the updating of the latent space **U** by equation

(4.10). In each iteration, the time complexity of updating $\mathbf{U}$ is $O(n^3)$, where n is the dimension of samples. Let T be the iteration number of MVP. The time complexity of MVP is $O(Tn^3)$

Convergence analysis For the convergence of MVP, it can be easily proved that the optimization problem in equation (4.8) can converge to a local optimum on the basis of the convergence analysis in [34]. We empirically find that the proposed MVP method converges rapidly, as shown in figure 4.2.

4.4 Applications to Multi-View Clustering

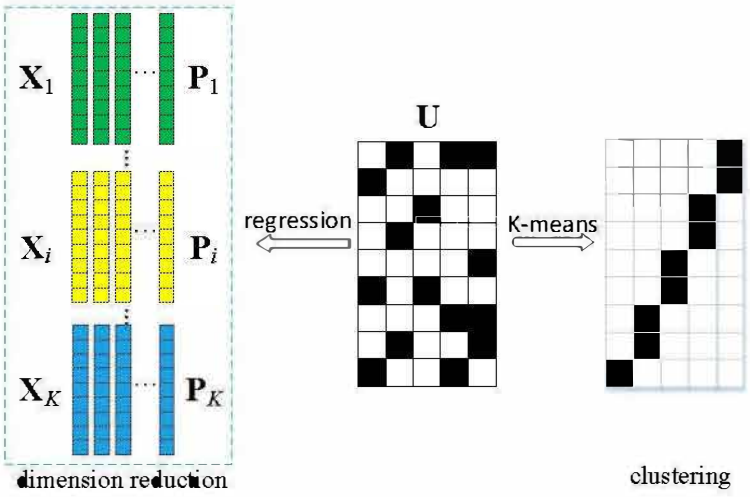


Figure 4.3. *Applications to multi-view clustering and unsupervised dimension reduction*

MVP learns a predictive latent representation for all views. Then we should consider how to use the latent space $\mathbf{U}$ for multi-view clustering and unsupervised dimension reduction.

MVP for clustering. We consider the latent space $\mathbf{U}$ as a new representation of multi-view data. Then we operate k-means algorithm on $\mathbf{U}$ to get the clustering labels, as shown in figure 4.3.

4.5 Experimental Results

In this section, we evaluate the proposed MVP on three benchmark datasets, and compare with the state-of-the-art multi-view unsupervised learning algorithms.

4.5.1 Datasets and Experimental Setup

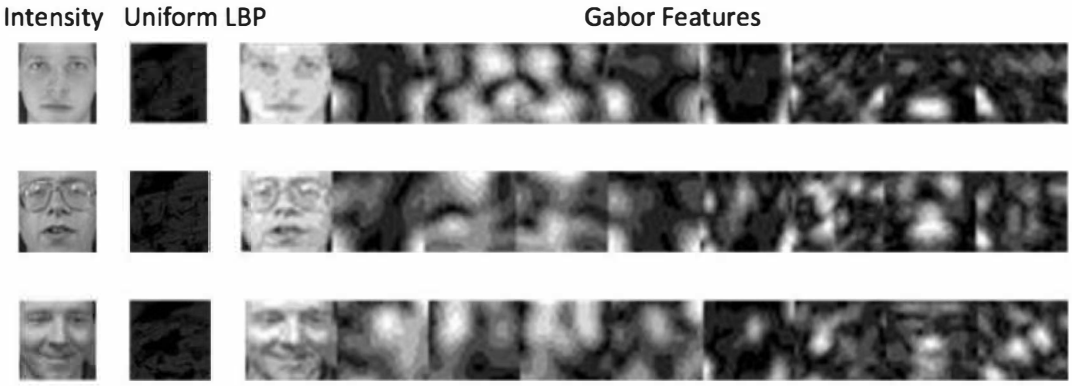


Figure 4.4. *The example images of ORL dataset.*

Datasets. We use three datasets (**BBCSport** dataset[11], **ORL** face dataset, **Corel800** dataset[12]) and the detailed descriptions about these datasets are given. 80% of samples are randomly selected for training and the rest for testing. Experiments are repeated for 10 times and the average result are reported.

BBCSport dataset [11] consists of 544 sports news articles in five classes (athletics, cricket, football, rugby, tennis). It is a synthetic multi-view dataset. Each document is segmented and segments are randomly assigned to the two views (3183 and 3203 dimensions). At most one segment from each document is assigned to the same view [51].

Corel800 dataset [26] contains 800 grayscale images of 10 individuals with 80 images per class. There are four types of feature, including LBP (59 dimensions) and GIST (512 dimensions) PHOG (680 dimensions), BOW feature (200 dimensions).

ORL face dataset, as already described in Chapter 3, this dataset contains 400 different images of 40 distinct subjects, which are taken at different times, under varied lighting conditions and facial expressions. We resize the image into 64x64, and extract three types of features: intensity (4096 dimensions), LBP (3304 dimensions) and Gabor (6750 dimensions).

Figure 4.4 presents 3 person's images in ORL face database with three types of features: intensity (4096 dimensions), LBP (3304 dimensions) and Gabor (6750 dimensions).

Evaluation metrics. We use one metric to evaluate the classification performance, and four evaluation metrics for clustering.

Table 4.1. *Clustering accuracy of multi-view clustering.*

Dataset	BBCSport	ORL	Corel800
SC(1)	0.367(0.000)	0.475(0.004)	0.332(0.002)
SC(2)	0.369(0.000)	0.453(0.002)	0.285(0.002)
SC(3)	—	0.391(0.004)	0.312(0.003)
SC(4)	—	—	0.233(0.002)
Co-regSC	0.318(0.022)	0.482(0.020)	0.316(0.010)
MMSC	0.360(0.002)	0.160(0.008)	0.146(0.002)
AMGL	0.426(0.020)	0.694(0.022)	0.305(0.026)
MVP	0.529(0.025)	0.714(0.037)	0.363(0.021)

Table 4.2. *NMI of multi-view clustering.*

Dataset	BBCSport	ORL	Corel800
SC(1)	0.023(0.002)	0.692(0.004)	0.209(0.002)
SC(2)	0.020(0.000)	0.665(0.003)	0.184(0.002)
SC(3)	—	0.582(0.005)	0.234(0.003)
SC(4)	—	—	0.118(0.002)
Co-regSC	0.018(0.002)	0.677(0.006)	0.219(0.017)
MMSC	0.010(0.002)	0.347(0.008)	0.021(0.003)
AMGL	0.110(0.016)	0.827(0.022)	0.247(0.026)
MVP	0.230(0.034)	0.867(0.020)	0.256(0.014)

- **Classification accuracy** is a classification quality evaluation measure. It is the percentage of the total number of data points that are correctly classified.
- **Clustering accuracy** is a simple and transparent evaluation measure. It gives the percentage of the clustering result.
- **Clustering purity** is a general measure of clustering. To compute purity, each cluster is assigned to the class which is the most frequent in the cluster. Then the accuracy of this assignment is measured by counting the number of correctly assigned data points.
- **NMI** normalizes the mutual information between the obtained clusters and the true clusters by the cluster entropies. NMI reaches its best value at 1 and worst at 0.
- **F-score** is a measure of test accuracy with ranges between 0 and 1. Higher values indicate closer match to the true clusters. It considers both precision and recall of the test stage.

Other than these metrics, we also use the Adjusted Rand Index (ARI) to evaluate the clustering results.

- **ARI** is a corrected-for-chance version of the rand index. The rand index is from 0 to 1 while the adjusted rand index is between -1 to 1.

Table 4.3. *Clustering purity of multi-view clustering*

Dataset	BBCSport	ORL	Corel800
SC(1)	0.373(0.001)	0.542(0.003)	0.343(0.002)
SC(2)	0.374(0.000)	0.516(0.003)	0.315(0.003)
SC(3)	—	0.473(0.005)	0.327(0.003)
SC(4)	—	—	0.250(0.002)
Co-regSC	0.367(0.002)	0.537(0.018)	0.339(0.010)
MMSC	0.361(0.002)	0.170(0.010)	0.151(0.003)
AMGL	0.443(0.017)	0.749(0.015)	0.353(0.021)
MVP	0.538(0.025)	0.765(0.031)	0.375(0.021)

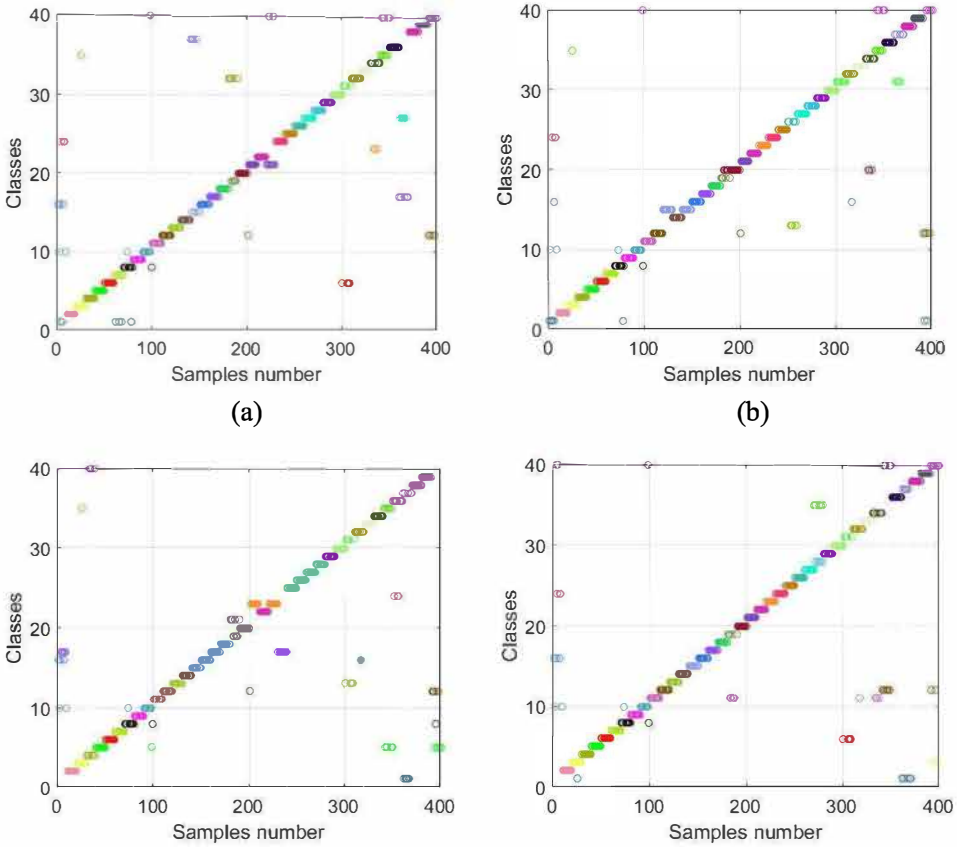


Figure 4.5. *Clustering results for ORL dataset on intensity, LBP, Gabor and the latent space* ***U****. (a) is the clustering result on intensity; (b) is the clustering result on LBP; (c) is the clustering result on Gabor; (d) is the clustering result on the latent space* ***U****. For all the four figures, the x-axis represents the number of each sample, and the y-axis represents the class of the samples have been clustered. Briefly speaking, if the sample on the diagonal line, it means the sample is clustering on a correct class;*

4.5.2 Experiments on Multi-View Clustering

For unsupervised clustering, we compare the proposed MVP with the following methods:

- **BSV:** The performance of the most informative view is reported, i.e., one that achieves the best spectral clustering performance[51].
- **Co-regSC:** The co-regularized multi-view spectral clustering [36] looks for clusters that are consistent across the different views.
- **MMSC:** The multi-modal spectral clustering [37] considers each type of feature as one modal, and learns a commonly shared graph Laplacian matrix by unifying different modals.
- **AMGL:** Auto-weighted multiple graph learning [39] reformulates the standard spectral clustering learning model by learning a weight for each graph automatically without introducing an additive parameter.

Table 4.4. *F-score of multi-view clustering.*

Dataset	BBCSport	ORL	Corel800
SC(1)	0.388(0.000)	0.341(0.005)	0.205(0.001)
SC(2)	0.384(0.000)	0.290(0.004)	0.183(0.001)
SC(3)	—	0.176(0.005)	0.208(0.002)
SC(4)	—	—	0.147(0.000)
Co-regSC	0.325(0.010)	0.308(0.016)	0.207(0.030)
MMSC	0.385(0.006)	0.034(0.020)	0.099(0.022)
AMGL	0.381(0.008)	0.509(0.036)	0.208(0.008)
MVP	0.440(0.012)	0.623(0.049)	0.246(0.010)

Table 4.5. *ARI of multi-view clustering.*

Dataset	BBCSport	ORL	Corel800
SC(1)	0.008(0.000)	0.321(0.006)	0.116(0.001)
SC(2)	0.003(0.000)	0.267(0.004)	0.092(0.001)
SC(3)	—	0.143(0.005)	0.118(0.002)
SC(4)	—	—	0.053(0.000)
Co-regSC	0.005(0.029)	0.287(0.041)	0.118(0.002)
MMSC	0.006(0.023)	0.003(0.044)	-0.001(0.014)
AMGL	0.034(0.033)	0.495(0.021)	0.098(0.000)
MVP	0.136(0.025)	0.615(0.051)	0.145(0.012)

There are two parameters for the proposed method, i.e., r and α in equation (4.8). We simply set r and α as 2 and 10 for all the datasets, respectively. For Co_regSC, we set the only parameter λ as 0.05, and the parameter r as 1 for MMSC according to

the experimental setting in [36] and [37]. The AMGL is a parameter-free method. Table 4.1, Table 4.2, Table 4.3, Table 4.4, and Table 4.5. show the clustering accuracy, normalized mutual information (NMI), clustering purity, F-score, and ARI on the three datasets, respectively. We can see the proposed method MVP is superior to all the competing methods.

To visually illustrate the effectiveness of the MVP method, we present the clustering results of ORL dataset in figure 4.5. As can be seen, compared to figure 4.5 (a-c), results in figure 4.5 (d) are more compact with most the points on the diagonal line. This means that, through the combination of the multi-view complementary information, Latent space provides more accurate clustering results than independent views.

4.6 Conclusion

To tackle the problem of unsupervised data, this chapter presented a multi-view predictable latent space learning (MVP) model and applied MVP to multi-view clustering and unsupervised dimension reduction. Compared with the existing multi-view unsupervised learning models, MVP emphasizes the predictability of the latent representation shared by multiple views. The predictability of the latent space is modeled by the correlation between the feature space and the latent space. To combine the multi-view complementary information, a weighted multi-graph is learned when the multi-view correlations are maximized. Experiments are conducted on three datasets with multiple features for face clustering. The results show that MVP achieves superior performances to the state-of-the-art algorithms. The main material of this chapter mainly comes from Publication III.

5. Robust Face Recognition via Sparse Representation of Gaussians

5.1 Introduction

Face recognition suffers from severe image blur, illumination variations and low resolution and other variations. In Chapter 2, Chapter 3 and Chapter 4, we present robust global and local feature descriptors for face images. In this Chapter, we present how to use high-order representation of features for robust face recognition.

High-order statistics of features are experimentally validated to bring about great performance gain for classification tasks [79][80]. Covariance descriptors have been widely used in different computer vision tasks, including face recognition [81], texture classification [82], action recognition [83]. Fisher vectors exploits first-order and second-order statistics and achieves superior performance to those using either zero-order or first order information, or their combination [84]. The first and second order central moments can be viewed as parameters of a Gaussian distribution. Hence, images or image sets can be represented by a collection of Gaussians [85]. Compared with covariance descriptor, Gaussian has additional mean information that has proven useful in [86].

Different from the vector space, a set of Gaussians form a Riemannian manifold. The covariance matrix is symmetric positive definite (SPD). By Gaussian embedding, a global Gaussian can be converted to a SPD matrix. Therefore, a collection of Gaussians forms a Riemannian manifold of SPD matrices. Distance metrics are defined or learned to measure the difference between two global Gaussians or two Gaussian Mixture Models [87]. Sparse coding and dictionary learning models are also developed for high-level feature extraction or classification [88]. The existing works focus on developing sparse coding models for Riemannian manifold. Note that for a Gaussian distribution, there are two components, i.e., mean vector and covariance matrix, which can be considered as two views for one sample. Hence, we can introduce multi-view learning to develop the classification models for Gaussians.

In this chapter, we propose a novel co-regularized sparse representation of Gaussians based classifier (CSRGC) [144]. An image can be modeled as a global Gaussian by extracting deep feature map. An image set can be modeled by a Gaussian distribution

as well. A query sample is first modelled as a global Gaussian and then represented jointly on the dictionary of mean vectors and the dictionary of covariance matrices. Experiments on face recognition show that the proposed CSRGC outperforms the state-of-the-art algorithms.

The rest of this chapter is organized as follows: Section 5.2 reviews the related work and Section 5.3 presents the proposed model. Section 5.4 conducts experiments and Section 5.5 concludes.

5.2 Related Work

In this section, we give a brief review of sparse representation based classification.

Sparse representation has been widely used in computer vision tasks, e.g., face recognition [89], image classification [90], visual tracking [91] and action recognition [92]. Sparse representation based classification represents a query sample on a dictionary composed of the training samples of all classes, and then classified by the reconstruction error of each class [89]. Additionally, the representation coefficients can be used as the extracted features for classification, e.g., linear spatial pyramid matching [90]. Most existing works focus on sparse coding and dictionary learning on the zero-order information, i.e., the original feature space. A negligible face that is the first-order and second-order statistics contain global information and take the correlation of the data into account. Therefore, the first-order and second-order statistics tends to be more robust to various variations in images and videos, e.g., variations of poses and illumination, occlusions.

Recently, researchers have proposed some sparse coding and dictionary learning models on Riemannian manifold. There are two widely used types of Riemannian manifold, i.e., Riemannian manifold of SPD matrices and Grassmann manifold. Sparse coding on Riemannian manifold is usually converted to a kernel sparse coding problem by deriving valid kernels for SPD manifold [93][94][95][96][97][98][88] or Grassmann manifold [99][100][101]. The covariance matrices for a collection of Gaussians can form a Riemannian manifold of SPD matrices. In addition, the global Gaussians can also form a SPD manifold by Gaussian embedding [102][103]. Hence, sparse coding of Gaussians also belong to coding on SPD manifold [85].

5.3 Co-regularized Sparse Representation of Gaussians

In this section, we present the proposed model, i.e., co-regularized sparse representation of Gaussians based classification (CSRGC).

5.3.1 Model

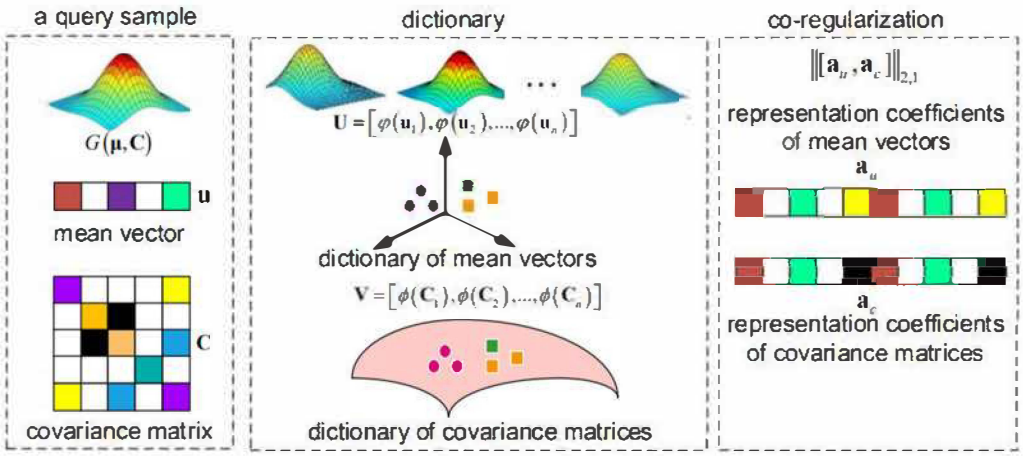


Figure 5.1. *The framework of co-regularized sparse representation of Gaussians based classification (CSRGC).*

For a group of image or image set data $\{X_1, X_2, \dots X_n\}$,we can first model their distribution with a Gaussian descriptor and get $G_i(u_i, C_i)$, i = 1,2, … , n,, where u_i is the mean value and C_i is the covariance matrix for the corresponding Gaussian descriptor G_i. As shown in figure 5.1, a query sample is modelled by a global Gaussian $\mathrm{G}(u, C)$. Then the co-regularized sparse representation of Gaussians based classification model is formulated as:

$$\begin{aligned}\min_{\{a_u,a_c\}}\{&w_1^r\|\varphi(u)-[\varphi(u_1),\varphi(u_2),\dots,\varphi(u_n)]a_u\|_2^2\\&+w_2^r\|\phi(C)-[\phi(C_1),\phi(C_2),\dots,\phi(C_n)]a_c\|_2^2+\lambda_1\|a_u\|_2^2+\lambda_2\|a_c\|_2^2\\&+\lambda_3\|[a_u,a_c]\|_{2,1}\}\end{aligned}$$

$$\text{s.t.}\, w_1+w_2=1, w_1\geq 0, w_2\geq 0 \qquad (5.1)$$

where w_1, w_2 are used to balance the weights of $\mathbf{u}$ and $\mathbf{C}$ of Gaussians, $\varphi(\cdot)$, $\phi(\cdot)$ are used to map $\mathbf{u}$ and C of Gaussians respectively. For mapping function $\phi(\cdot)$, we can use different kinds of mapping functions for the covariance matrix. Here, $\varphi(u) =$

u and $\phi(u) = vec(log(\mathbf{u}))$, where $log(cot)$ is the logarithm operation and $vec(\cdot)$ is the vectorization operation.

5.3.2 Optimization and Algorithm

For the objective function in equation (5.1), it is generally non-convex. We use alternation minimization to solve the optimization problem. Each sub-problem of equation (5.1) is convex. We use some symbols to simplify the objective function,

$$\mathbf{U} = [\varphi(\mathbf{u}_1), \varphi(\mathbf{u}_2), \dots, \varphi(\mathbf{u}_n)]$$
$$\mathbf{V} = [\phi(\mathbf{C}_1), \phi(\mathbf{C}_2), \dots, \phi(\mathbf{C}_n)]$$
$$\mathrm{b} = [\mathrm{a_u}, \mathrm{a_c}]$$

where **U** is the dictionary of mean vector, **V** is the dictionary of covariance matrices.

Update $\mathbf{a_u}, \mathbf{a_c}$: The partial derivatives of objective function with respect to $\mathbf{a_u}$, $\mathbf{a_c}$ will be set to 0.

$$\frac{\partial f}{\partial \mathbf{a_u}} = w_1^r(-2\mathbf{U}^T\varphi(\boldsymbol{u}) + 2\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{a_u}) + 2\lambda_1\mathbf{I} + 2\lambda_3\mathbf{G} = 0 \tag{5.2}$$

$$\frac{\partial f}{\partial \mathrm{a_c}} = w_2^r(-2\mathbf{V}^T\phi(\mathbf{C}) + 2\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{a_c}) + 2\lambda_2\mathbf{I} + 2\lambda_3\mathbf{G} = 0 \tag{5.3}$$

where **G** is a diagonal matrix with the i^{th} diagonal element as $\frac{1}{2\|b_i\|_2}$.

Thus we can get the solution of $\mathbf{a_u}$, $\mathbf{a_c}$ in each iteration.

$$\mathbf{a_u} = \left(\mathbf{U}^T\mathbf{U} + \frac{\lambda_1}{w_1^r}\mathbf{I} + \frac{\lambda_3}{w_1^r}\mathbf{G}\right)^{-1}\mathbf{U}^T\varphi(\mathbf{u}) \tag{5.4}$$

$$\mathbf{a_c} = \left(\mathbf{V}^T\mathbf{V} + \frac{\lambda_2}{w_2^r}\mathbf{I} + \frac{\lambda_3}{w_2^r}\mathbf{G}\right)^{-1}\mathbf{V}^T\phi(\mathbf{C}) \tag{5.5}$$

Update w: The parameter r is set to 2, thus this optimization problem can be transformed into a quadratic programming problem and we can get a closed form solution for w_1, w_2.

Update G: G can be updated as follows:

$$g_i = \frac{1}{2\|[a_{ui}, a_{ci}]\|_2 + eps} \tag{5.6}$$

According to [104], for a general convex problem, the alternating minimization approach would converge to the correct solution. As the sub problems of equation (5.1) are all convex, the optimization problem in equation (5.1) can converge.

5.3.3 Classification

By co-regularized sparse representation of Gaussians, we can get the representation coefficients vectors that correspond to mean vector and covariance matrix, respectively. The query data can be classified according to the weighted reconstruction error of each class.

$$l(\mathbf{X}) = \arg\min\{e_i\}$$
$$e_i = \left\{ \begin{array}{l} w_1^r \left\| \varphi(\mathbf{u}) - [\varphi(\mathbf{u}_{i_1}), \varphi(\mathbf{u}_{i_2}), ..., \varphi(\mathbf{u}_{i_k})]\mathbf{a}_{\mathbf{u}}^i \right\|_2^2 + \\ w_2^r \left\| \phi(\mathbf{C}) - [\phi(\mathbf{C}_{i_1}), \phi(\mathbf{C}_{i_2}), ..., \phi(\mathbf{C}_{i_k})]\mathbf{a}_{\mathbf{c}}^i \right\|_2^2 \end{array} \right\} \tag{5.7}$$

where e_i is the reconstruction error of the i^{th} class.

5.4 Experimental Results

In this section, experiments are conducted on video based face recognition and image set classification to validate the performance of the proposed algorithm.

5.4.1 Datasets

We conduct experiments on two face databases, including YouTube Celebrities [106] and LFW [110]. The samples of these two databases are shown in figure 5.2.

Figure 5.2. *The example images of two face databases.*

YouTube Celebrities (YTC) database contains 1910 video clips of 47 subjects [106] with different numbers of frames in each video. Following [108], [109], we use histogram equalization to eliminate effects of light in pre-processing step and randomly select 3 videos per subject for gallery and 6 videos for probes. Then, each image is resized to a 20x20 image with the intensity feature and each video can be

expressed by the matrix of $n_i \times 400$ where n_i is the number of frames in each video.

The LFW [110] database consists of images of 5,749 individuals in the wild environment [110]. LFW-a is one aligned version of LFW. Each face is aligned by using a commercial face alignment software [111]. We gathered the face image of 158 subjects the number of whose face images is more than 10 from LFW-a. Half of the face images per subject is used for training and the rest for testing. Random experiments are run 10 times and the average performance is reported.

5.4.2 Comparison Methods

To illustrate the efficiency of the proposed model, we compare our method CSRGC with the state-of-the-art non-linear learning methods. They are summarized as follows.

- Nonlinear manifold based methods: Manifold-Manifold Distance (MMD) [112]; Manifold Discriminant Analysis (MDA) [113].
- Affine subspace based methods: Affine Hull based Image Set Distance (AHISD) [114]; Convex Hull based Image Set Distance (CHISD) [114].
- AIRM and Stein based methods: SPD Manifold Learning (SPDML) [115].
- SPD manifold based methods: Covariance Discriminative Learning (CDL) [108]; Log-Euclidean Metric Learning (LEML) [109].

5.4.3 Parameter Setting

For fair comparison, we exploit the source codes of comparison methods provided by the authors, and set the parameters suggested by the corresponding papers. For MMD, the PCA percentage is set to 90%. For MDA, we set the number of local models, between-class NN local models and the subspace dimension the same as [113]. For SPDML, we implement both SPDML-AIRM and SPDML-Stein versions. In both versions, following [115], v_w is set as the minimum of the samples in one class. The new dimensionality of the low-dimensional manifold and v_b are tuned by 5-fold cross-validation. We compare CSRGC with both linear and non-linear versions of AHISD and CHISD [114], where 98% energy by PCA is retained in non-linear AHISD and the value of error penalty C in CHISD is set as same as [114]. For CDL,

the distance metric is learned with linear discriminant analysis (LDA) and partial least squares (PLS) in Hilbert space. The reduced feature dimension is set to (c-1) for LDA, where c is the number of classes. For LEML, is tuned from 0.001 to 1000 and the value of is tuned from 0.1 to 1.

There are seven parameters $w_1, w_2, r, \lambda_1, \lambda_2, \lambda_3, n$ for C-SRGC. We set the initial value of $w_1 = w_2 = 0.5$, r is fixed to 2, the number of iterations is set to 50 and all regularization parameters $\lambda_1, \lambda_2, \lambda_3$ are all set to 0.001.

5.4.4 Experimental Analysis

The experimental results on YTC and LFW datasets [110] are listed in Table 5.1. The results show that the proposed CSRGC outperforms the competition methods, including affine subspace based methods, nonlinear manifold based methods, and SPD manifold based methods. The superior performance of CSRGC benefits from the following aspects. Firstly, Gaussian descriptors capture the high-order statistics of data, which is more robust the image or video variations. Secondly, sparse representation of Gaussians uses the information of all classes compared with distance based classifiers. Thirdly, the weights take the importance of the first order and second order information into account.

Table 5.1. *Recognition accuracy on YouTube Celebrities and LFW databases*

Methods	YTC	LFW
MMD	69.60	34.81
MDA	64.72	38.52
AHISD(linear)	64.65	25.10
AHISD(non-linear)	66.58	26.90
CHISD(linear)	67.24	25.32
CHISD(non-linear)	68.09	27.22
SPDML-AIRM	67.50	62.58
SPDML-Stein	68.10	63.64
CDL-LDA	70.21	71.55
LEML	69.85	68.72
CSRGC	80.53	72.85

Figure 5.3 shows the query faces misclassified by the proposed model. The first column shows the query samples. The other columns list the samples of classes that CSRGC classifies the query sample to. The labels of the face images in the red bounding boxes

are different from the query sample while the one in the green bounding box share the same label with the query sample. As there are great pose and illumination variations, the query samples are easily misclassified. Hence, more robust features and classifiers are needed for face recognition in the wild.

Figure 5.3. *Five query samples misclassified by the proposed method.*

5.5 Conclusions

To tackle the problems of severe image blur, illumination variations, low resolution and other variations in face recognition, this chapter presented a co-regularized sparse representation of Gaussians based classification model (CSRGC) for pattern classification. To make full use of the components of a Gaussian distribution, a query sample is classified by joint representation of mean vector and covariance matrix. A co-regularization is imposed on the representation coefficient vectors of both mean vector and covariance matrix. Experiments on material classification and face recognition show that the proposed CSRGC achieves superior performance in comparison with the state-of-the-art algorithms. The main material of this chapter mainly comes from Publication IV.

6. Robust Deep Face Recognition with Noisy Labels

6.1 Introduction

Deep learning has achieved consistent breakthroughs in face recognition [119]. The superior performance of deep learning owes to the representations of data with multiple levels of abstraction and massive labelled training data [116]. However, the lack of accurate label information makes it hard to learn a well-trained deep model with only a few labelled samples. For face recognition, despite the success of deep learning in face verification [117][118], it is hard to achieve satisfactory recognition accuracy without sufficient training data, especially when there are a large number of subjects in face identification. DeepFace uses a large-scale face dataset that consists of 4 millions face images of 4000 subjects [79]. FaceNet is learned on a much larger dataset with 200 million images with 8 million subjects [117]. The large-scale face databases with accurate labels dramatically improve the performance of face recognition in that the deep learning models can be well trained.

In real-world applications, the collected data are mixed with severe label noise, which significantly degrades the generalization ability of deep learning models. How to acquire a clean face dataset (faces are labeled correctly) is one of the key challenges in face recognition. One intuitive way is to manually collect and label the face images. The other way is a semi-automatic annotation through online image searching, which manually corrects label noises in the automatic online-searching results. Clearly, the two ways mentioned above suffer from high time consumption, high labelling expense and inevitable labelling error [119]. Hence, there is an urgent need to construct an effective face tagging method that can automatically remove labelling noises, thus enable the collection of a large-scale face dataset with accurate identification information.

In this chapter, we introduce a robust deep face recognition method which alleviates the impact of the label noise through automatic outlier removal [145]. Through the automatic dataset cleanup process (automatically remove the faces with label noise), the performance of learned deep models can be boosted. Experiments on large-scale face datasets LFW [110], CCFD, and COX [16] show that RDFR can effectively remove the label noise and improve the face recognition performance.

The rest of this chapter is arranged as follows: In Section 6.2 we give a brief review of related works. In Section 6.3, we propose Robust Deep Face Recognition method. Section 6.4 gives the experimental results. We then conclude in Section 6.5.

6.2 Related Works

There are mainly three types of methods dealing with label noise: noise-robust, noise-removal, and noise-tolerant. The first category of methods seeks ways to learn models which are robust to label noise. Manwani et. al proposed in [120] that given the loss functions, the learned model is robust to noise if the misclassification probability is irrelevant to label noise. Patrini et al. proposed to improve label noise robustness by loss factorization in weakly supervised learning [121]. Gao et al. used risk minimization to divide the loss function into two parts: one irrelevant to noise and the other related to noise [122]. The second category of methods seeks ways to relabel or directly discard noisy face images. These methods need to set a threshold manually for noise removal [123]. Wilson et al. reviewed the noise removal methods based on locality smoothness. Brodley et al. proposed to detect noisy samples through the setting of classification confidence scores [124]. The third category of methods seeks to model the noise distribution, that directly separates the classification model from the noise model. The most common noise modeling method is to estimate the noise distribution by the Bayesian methods.

For face recognition, noise removal aims to clean the noisy samples of each subject and then get a clean face dataset. Other than visual information, the side information can help to correct the label noise. Schroff et al. proposed to fuse visual and textual information to reorder the face images [125]. Li proposed to reorder the samples by incremental model learning using the searching results as the initialized rank [126]. In real-world applications, the small-scale manually labelled face dataset and the side information can be not reliable [127]. Hence, it is one of the most challenging issues to automatically detect noise samples in unsupervised setting and develop robust deep face recognition model.

6.3 Robust Deep Face Recognition

6.3.1 Framework

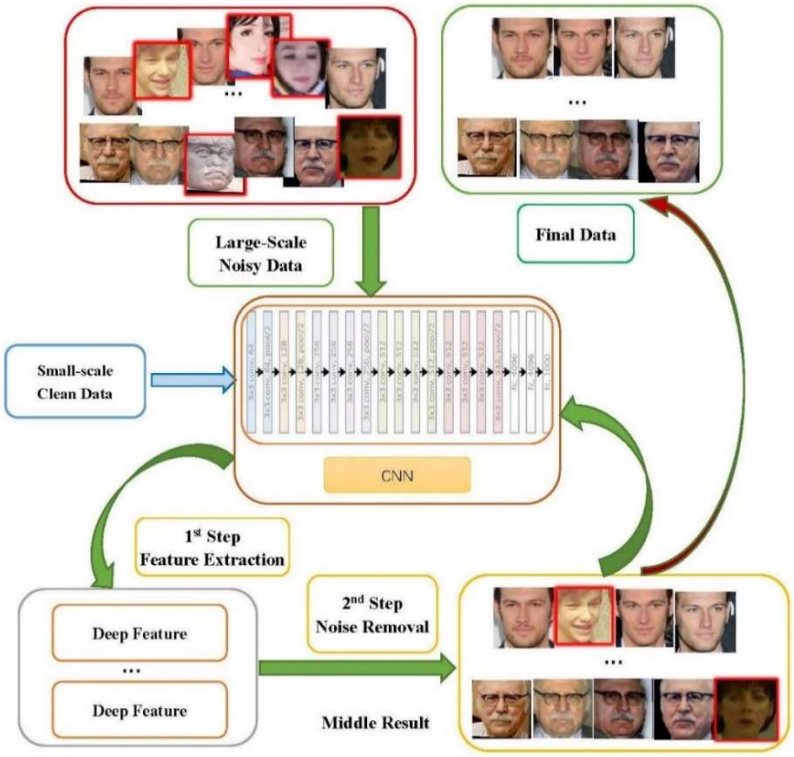


Figure 6.1. *The flowchart of robust deep face recognition via automatic label noise removal.*

To make the best use of the large-scale noisy data, we propose a robust deep face recognition (RDFR) method with automatic noise removal. The framework is given in figure 6.1. Firstly, a deep model is trained on a clean dataset with a small sample size. The deep features are extracted for a large-scale noisy face dataset using the pre-trained deep model. Then, the noisy samples are removed by unsupervised one class learning (UOCL). The above process is repeated until the recognition rate on the validation set does not increase. RDFS aims to extract a clean subset from the large-scale noisy data to train a better deep model.

The noise removal stage plays a dominant role in the proposed model in that in this chapter we focus on the impact of the label noise on face recognition. By noise removal, more robust and discriminative features can be extracted by the learned deep model. The large amounts of clean data after noise removal can also be used to boost the performance of the deep models.

6.3.2 Unsupervised One Class Learning (UOCL)

In real-world applications, face images are easy to acquire, which gives the possibility of retrieving large-scale datasets. Yet the dataset could be quite noisy, severe outliers should be removed from the large-scale dataset to make the visual information of face images well utilized. The common strategy to deal with a label noise is to transform outlier removal to an unsupervised one-class learning task. The representative methods are robust kernel density estimation (RKDE) [128] and sparse modeling for finding representative objects (SMRS) [129]. In this work, we introduce an efficient automatic noise removal method, namely, Unsupervised One Class Learning (UOCL) [130]. UOCL is built upon two intuitive assumptions: 1) outliers originate from low-density samples, and 2) neighboring samples tend to have consistent classifications.

Given an unlabeled dataset $\chi = \{x_i \in \mathbb{R}^d\}_{i=1}^n$ we aim to get a classification function $f: \mathbb{R}^d \mapsto \mathbb{R}$, which is similar to one class SVM. By leveraging a kernel function $\kappa: \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ that induces the Reproducing Kernel Hilbert Space (RKHS) the target classification function is in the following expression:

$$f(x) = \sum_{i=1}^n \kappa(x, x_i)\alpha_i \tag{6.1}$$

where α_i is the expansion coefficient contributed by the functional base $\kappa(., x_i)$. Let us introduce a soft label assignment $\mathcal{Y} = \{y_i \in \{c^+, c^-\}\}_{i=1}^n$, where c^+ is a positive value for positive samples and c^- is a negative value for outliers. Let $\mathrm{y} = [y_1, \cdots, y_n]^T$ be the vector representation of $\mathcal{Y}$.

Now we establish the UOCL model as minimizing the following objective:

$$\min_{f \in \mathcal{H}, \{y_i\}} \sum_{i=1}^n (f(x_i) - y_i)^2 + \gamma_1 \|f\|_M^2 - \frac{2\gamma_2}{n^+} \sum_{i, y_i > 0} f(x_i) \tag{6.2}$$

$$s.t.\ y_i \in \{c^+, c^-\}, \forall i \in [1:n], 0 < n^+ = |\{i | y_i > 0\}| < n$$

where $\gamma_1, \gamma_2 > 0$ are two trade-off parameters controlling the model, $\|f\|_M^2$ is the manifold regularization item.

6.3.3 Deep Model

For label noise removal, we use VIPLFaceNet [132] and in the stage of face recognition, we use Resnet-VIPL [132]. VIPLFaceNet contains 7 convolution layers and 3 full connected layers. Resnet-VIPL is modified from the classic Resnet [131], and consists

of 82 convolution layers and 2 full connected layers. Compared with Resnet-101, Resnet-VIPL greatly reduces the computation burden while keeps the performance.

6.4 Experimental Results

Experiments are conducted on large-scale face databases to evaluate the performance of the proposed method.

6.4.1 Datasets

We use a large-scale noisy face dataset MS-Celeb-1M [14] together with a small-scale clean dataset CASIA-WebFace [10] for training. The performance is evaluated on three datasets, including LFW [110], CCFD [15] and COX [16].

MS-Celeb-1M [14] is a large-scale noisy dataset from Microsoft. MS dataset [14] has 8,456,240 real-world facial images of 99,891 identities. It is a large-scale dataset that contains large variations in age, pose and so on. There are severe label noises, which may degrade the performance of deep models.

CASIA-WebFace Database [10], published by Chinese Academy of Sciences in 2014, is a semi-automatically collected dataset. The face images in this dataset are from internet and so covers a large scale of pose and illumination variations. There are 10575 subjects and 494414 face images in this dataset.

COX [16] consists of the gallery set and probe set. The gallery set contains 20,312 face images of 20,312 subjects. The images in the gallery set are the face images of the Chinese identity card. The probe set contains 1,102 test images, which are collected in the wild.

CCFD [15] (Chinese Celebrity Face Dataset) is a large-scale real-world face dataset collected by VIPL. This dataset consists of 263,696 images of 1,001 subjects, with two subsets for training and testing. The training set contains 171,792 images of 701 subjects and the test set contains 91,904 images of 301 subjects. Facial images in CCFD are collected in real-world environments from the internet and has large variations in age, expression, light, occlusion and pose.

The test protocols of the three datasets are as follows:

1. For LFW dataset, the average face verification rate of ten folds are used, where in each fold there are 300 positive pairs and 300 negative samples.

2. For CCFD dataset, the verification rate under different false acceptance rate is used to evaluate the recognition performance. Here, the verification rate when FAR is 0.1 is reported.
3. As for COX dataset, the ROC curse is used to evaluate the performance.

6.4.2 Parameter Settings

Face preprocessing. The face images of different datasets are all resized to 256× 256. Deep features are extracted for label noise removal and face recognition. The deep feature dimension for noise removal is 2,048 and the dimension for face recognition is 1,024.

Parameter setting. All our experiments are based on Caffe platform. SGD is utilized to train the VIPLFaceNet and Resnet-VIPL. For VIPLFaceNet, we set the base learning rate as 0.06, mini-batch size as 128, iteration size as 1, total iteration in pre-train process as 120,000, momentum as 0.9, and weight-decay as 0.0002. The learning rate is decreased according to the polynomial policy with gamma value equals to 0.5. For Resnet-VIPL, we set the base learning rate as 0.04, mini-batch size as 32, iteration size as 4, total iteration in pre-train process as 300,000, momentum as 0.9, and weight-decay as 0.0002. For UOCL, we use Gaussian kernel $\mathrm{k}(x, y) = exp(-\|x - y\|^2)/2\sigma^2$, where $\sigma = \sum_{i,j=1}^{n} \frac{\|x_i - y_i\|^2}{n^2}$.

6.4.3 Experimental Analysis

We use CASIA-WebFace [10] as the clean dataset with small sample size to train a CNN model. Noisy samples are iteratively removed from MS-Celeb-1M [14]. Comparison is made between the recognition rate of raw noisy dataset and the cleaned dataset.

Figure 6.2. shows the process of label noise removal on MS-Celeb-1M database. Red bounding box represents the correctly labelled samples while green bounding box represents noisy samples. It can be seen from figure 6.2 and figure 6.3 that noise removal process may also discard some clean samples while removing all noisy samples.

The deep features learned in this work are used for recognition rather than some low-level applications, e.g., semantic segmentation, saliency detection. Therefore, the models extract high-level semantic features that are quite useful for face recognition.

Compared with the classical approaches, for deep learning methods, feature extraction and classifier learning are jointly conducted, which can extract more discriminative features with low redundancy.

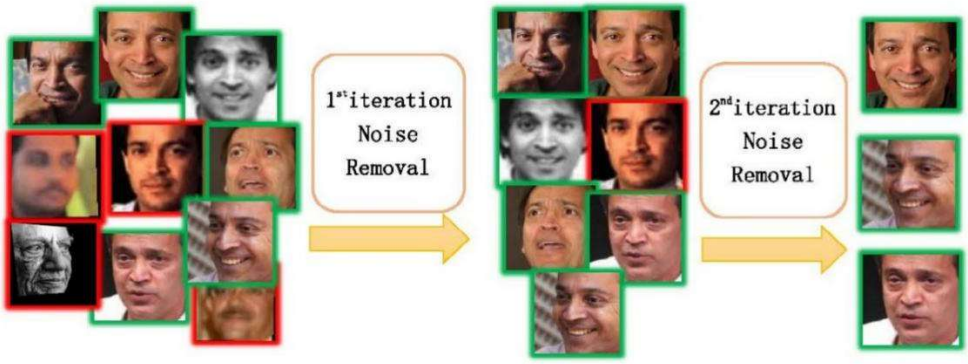


Figure 6.2. *The process of label noise removal on MS-Celeb-1M database. Red bounding box represents the correctly labelled samples while green bounding box represents noisy samples. The face images of one person is taken as example.*

Table 6.1. *The face verification rate on LFW dataset*

Method	Training dataset	Accuracy
Resnet-VIPL	MS-Celeb-1M	99.25%
Resnet-VIPL	MN_01	99.40%
Resnet-VIPL	MN_02	99.25%
DeepFace	SFC	97.35%
WSTFusion	WSTFusion	98.73%
VGGFace	VGGFace	98.95%
DeepID2+	DeepID2+	99.47%
FaceNet	Google	99.63%

Table 6.1 shows the face verification rate on LFW dataset. Suffix MN_01 and MN_02 represent the results after the fist and the second noise removal respectively. The results show that compared with the raw noisy data, the verification rate is improved by 0.25% after the first-iteration noise removal. Compared with DeepFace, VGGFace and DeepID2+, the performance of the proposed method is superior or comparable.

It should be noted that FaceNet achieves 99.63% in that it uses 200 million face images to train the deep model. It should also be noted that although the recognition rate is the same as the raw noisy data after noise-removal after second iteration. However, the number of training samples is only a quarter of the raw data. Hence, the time consumption and storage burden is greatly reduced.

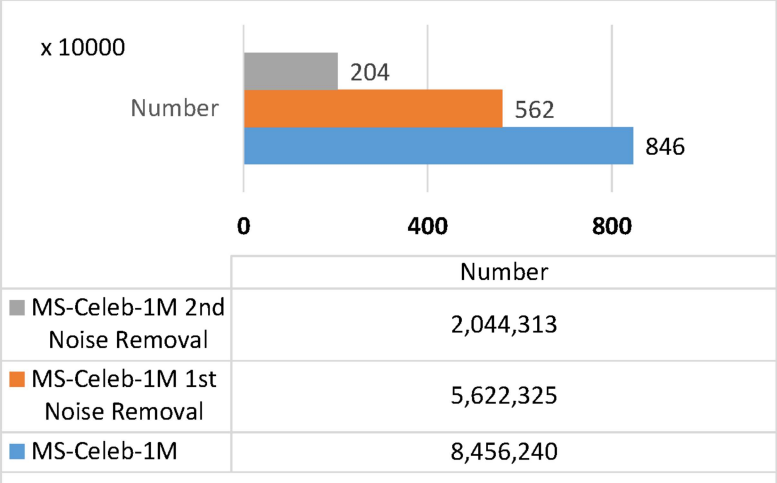


Figure 6.3. The number of face samples in MS-Celeb-1M before and after noise removal.

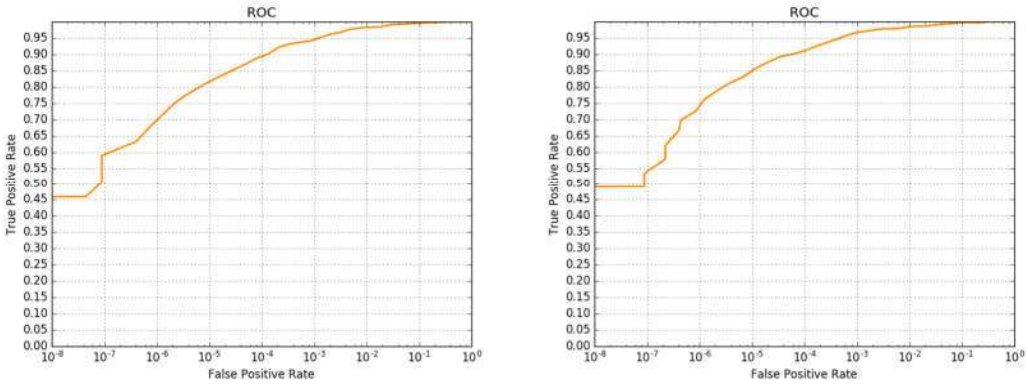


Figure 6.4. *The comparison of ROC curves on COX dataset. Left: The ROC curve when MS-Celeb-1M is used for training. Right: The ROC curve when the cleaned MS-Celeb-1M is used for training*

Table 6.2. *The face recognition rate on CCFD dataset*

Method	**Training dataset**	**Fine-tune**	**Accuracy**
Resnet-VIPL	MS-Celeb-1M	No	58.10%
Resnet-VIPL	MN_01	No	64.72%
Resnet-VIPL	MN_02	No	61.19%
Resnet-VIPL	MS-Celeb-1M	Yes	65.04%
Resnet-VIPL	MN_01	Yes	70.66%
Resnet-VIPL	MN_02	Yes	68.41%

Table 6.2. shows the recognition rate on CCFD dataset [15]. Note, that the face images in MS-Celeb-1M are all collected from European and American while CCFD contains only the face images of Chinese Celebrities. To narrow the gap across different ethnic

groups, we fine-tune the parameters on the training set of CCFD to improve the recognition performance. From the result, we can see that similar to LFW, the model trained on MN_01 is much better than on MS-Celeb-1M. Compared with the result without fine-tuning, the recognition rate is much improved. Note that after the second noise removal, the rate slightly decreases, since too many clean data have been removed together with the noisy face images. Figure 6.4. shows results on COX dataset [16]. The ROC curves before and after label noise removal clearly reflect the effectiveness of the proposed method.

6.5 Conclusions

To tackle the problem of label noise in datasets, this chapter presented a robust deep face recognition method by automatically noise removal. Unsupervised one-class learning is used to remove the massive noisy face images. Experiments on large-scale face datasets in the wild validate the effectiveness of the proposed method. The main material of this chapter mainly comes from Publication V.

7. Multi-task Deep Face Recognition for Insufficient Dataset

7.1 Introduction

Other than label noise in massive dataset, there are also challenges when the dataset have insufficient samples. Thus in this chapter, we will discuss how to generate deep learning methods in case of small datasets. The main idea here is to transfer tasks on other large-scale dataset to current task on a limited-scale dataset. Motivated by multi-task learning, learning on multiple datasets can also be assumed as a multi-task learning problem. Thus we propose a multi-task deep learning method for face recognition using multiple face datasets [146]. Through the sharing of common representation layers among datasets, the proposed method can achieve better performance than straightforward fine-tuning. Besides, the accuracy on the initial task also can be maintained. Experiments on LFW [110] and CCFD [15] validate the effectiveness of the proposed multi-task deep learning model for face recognition.

The rest of this chapter is organized as follows: Section 7.2 presents the multi-task deep learning framework. Section 7.3 introduces multi-task deep face recognition with different face datasets. Section 7.4 conducts experiments. Section 7.5 concludes.

7.2 Multi-task Learning Model

7.2.1 Task Loss

Figure 7.1. shows the structure of the multi-task deep learning method. To train a deep model using multiple face datasets, each dataset is considered as an individual task. The datasets share common representation with dataset-specific fully connected layers.

Assume that there are two classification tasks: Task A has C_A series and task B has C_B series. In Caffe [78], for each mini-batch, the loss function of task A:

$$L_A = \frac{-1}{N_A}\sum_{n=1}^{N_A}\log(\hat{p}_n l_n),\ \ l_n \in [0,1,\dots C_A - 1] \tag{7.1}$$

where N_A is the images number in one mini-batch and $\hat{p}_n l_n = \frac{e^{x_n l_n}}{\sum_{c=0}^{C_A-1} e^{x_n c}}$. Thus, the loss of two tasks is given as follow:

$$L_A = \frac{-1}{N_A}\sum_{n=1}^{N_A} log\left(\frac{e^{x_n l_n}}{\sum_{c=0}^{C_A-1} e^{x_n c}}\right), l_n \in [0,1, \dots C_A - 1] \tag{7.2}$$

$$L_B = \frac{-1}{N_B}\sum_{n=1}^{N_B} log\left(\frac{e^{x_n l_n}}{\sum_{c=0}^{C_B-1} e^{x_n c}}\right), l_n \in [0,1, \dots C_B - 1] \tag{7.3}$$

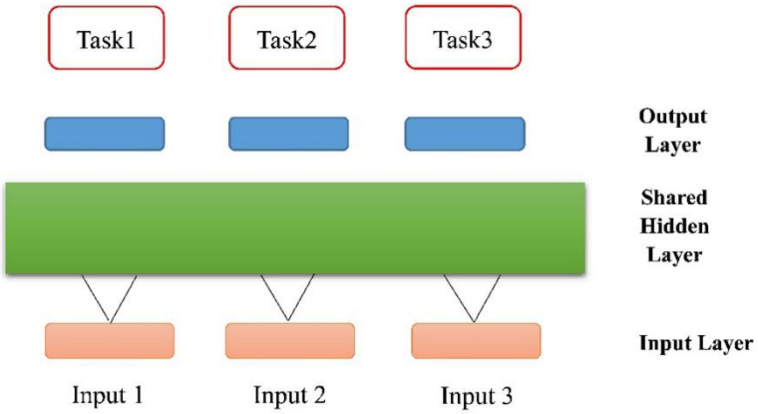


Figure 7.1. *Deep network for multi-task deep learning. Input 1, Input 2 and Input 3 represent data of different tasks. The green part represents shared hidden layers of multi-task deep learning network. The blue part represents unique output layers of different tasks. Task 1, Task 2 and Task 3 represent the loss function for each task. The proposed multi-task deep learning model learns shared hidden layers while designing task-specific output layers.*

7.2.2 Back Propagation

As shown in figure 7.1, each task calculates its own loss. Whereas, during back propagation, all gradients will be added to update the parameters of the deep model. Multi-task deep learning enables us to obtain knowledge from other tasks through shared representation, thus avoids overfitting in single task.

7.3 Multi-task Deep Learning for Face Recognition

The framework of Multi-task deep learning is shown in figure 7.2. The model is transferred from a large-scale Dataset A to a limited-scale Dataset B.

Step 1: Generate a pre-trained model using Dataset A.

Step 2: Dataset B together with Dataset A are used to fine-tune the pre-trained model. The datasets are united in axis n as shown in figure 7.3, with labels united. And the combined data are used to train the convolutional neural network.

Step 3: Multi-task deep learning on multiple datasets. Generally, the penultimate fully connected layer is concerning feature representation while the last fully connected layer is for classification work. Thus, the data split after the penultimate fully connected layer before classification. Then every task calculates their own loss. The structure of data split is shown in figure 7.3. In order to be consistent with the uniting process, data split also at the axis of n. All above operations are on the sample space and their labels will be fixed.

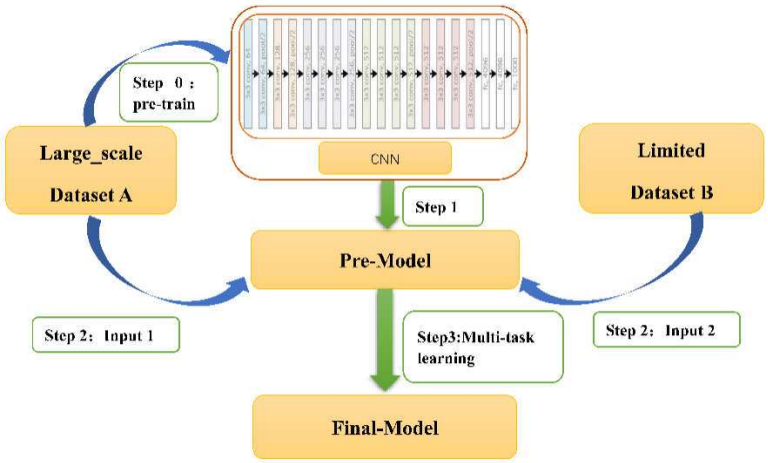


Figure 7.2. *Overview of Multi-task deep learning.*

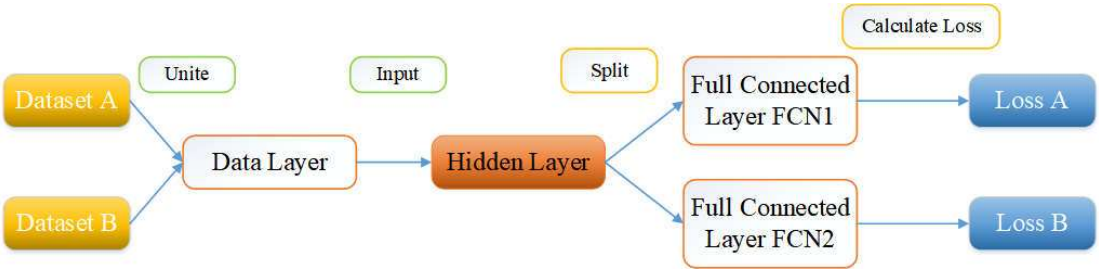


Figure 7.3. *Dataset concatenation and split.*

7.4 Experimental Results

7.4.1 Dataset

Multi-task deep learning method is evaluated on the real-world datasets: MS dataset [14], CCFD [15] dataset and LFW [110] dataset.

MS dataset [14] is a subset of large-scale noisy dataset of real-world called MS-Celeb-1M [14]. After artificially marked, MS dataset is clean enough. MS dataset has 3,095,536 real-world facial images of 41,857 identities totally. It is a large-scale real-world dataset that contains large variations in age, pose and so on.

MS-Celeb-1M [14] is a large-scale noisy dataset from Microsoft. MS dataset [14] has 8,456,240 real-world facial images of 99,891 identities. It is a large-scale dataset that contains large variations in age, pose and so on. There are severe label noises, which may degrade the performance of deep models.

In order to ensure the consistency of the experiments, all the face images are normalized to 256 × 256 and aligned using five facial points. The faces in MS dataset are mostly from European and American, while the identities in CCFD dataset are all Chinese. The detection of these two datasets can be assumed as two different tasks due to the ethnic difference, thus meets our assumption of multi-task deep learning.

7.4.2 Parameter Settings for CNN

Same Resnet-VIPL network is used as convolution neural network to train deep model, as described in Section 6.4. All experiments are based on Caffe platform. SGD is utilized to train Resnet-VIPL. We set the base learning rate as 0.04, mini-batch size as 32, iteration size as 4, total iteration in pre-train process as 300K, iteration in multi-task deep learning as 320K, iteration in direct fine-tune as 80K, power as 0.5, momentum as 0.9, weight-decat as 0.0002. The learning rate decreases according to the polynomial policy with γ value equal to 0.5. The base learning rate of direct learning is 0.0005 and the base learning rate of multi-task deep learning method while fine-tuning is 0.008. The dimension of face feature is 1,024.

7.4.3 Experimental Analysis

The test set of CCFD [15] contains 91,904 images of 301 subjects and then it was divided into two parts named Target set and Query set. We chose 50% images from every identity in CCFD test set randomly as Query set and the rest images are as Target set. We evaluate accuracy according to similarity matrix $Sim_{(i,j)}$, which represents the similarity between the i^{th} image in Query set and the j^{th} image in Target set. The

performance is evaluated using verification rates when false acceptance rate equals 0.1%.

Table 7.1 and Table 7.2 gives the comparison results of verification accuracy on LFW and CCFD datasets respectively. It can be seen that multi-task deep learning methods achieves better performance than direct fine-tuning on both datasets. Although multi-task learning slightly dropped (-0.1%) on LFW dataset compared with pre-trained model, the improvement is huge (7.6%) on CCFD dataset. This improvement can be attributed to the fact that the proposed method makes full use of large-scale dataset through multi-task learning and transferring.

Table 7.1. *The accuracy on LFW*

Deep method	Dataset of pre-train	Learning method	Accuracy
Resnet-VIPL	MS	None	99.23%
Resnet-VIPL	MS	Fine-tune	98.38%
Resnet-VIPL	MS	Multi-task	99.13%

Note: The value of FAR is set as 10% on LFW and 0.1% on CCFD.

Table 7.2. *The accuracy on CCFD*

Deep method	Dataset of pre-train	Learning method	Accuracy
Resnet-VIPL	MS	None	64.28%
Resnet-VIPL	MS	Fine-tune	69.46%
Resnet-VIPL	MS	Multi-task	71.88%

7.5 Conclusions

To tackle the problem of insufficient dataset for a particular task, this chapter explored the possibility of transferring tasks on large-scale datasets to tasks on a limited-scale dataset. We present a multi-task deep learning method for face recognition. Datasets with diverse properties are considered as different tasks. Through the iterative combination of pre-training process and fine-tuning process, the proposed method makes better use of the large-scale dataset. Experiments on LFW and CCFD datasets show that MTDL generalizes well on different face databases through using large-scale dataset as supplement to the limited-scale dataset. The main material of this chapter mainly comes from Publication VI. To further boost the performance of small sample problems, we will consider data augmentation and multi-task learning strategy together in the future.

8. Conclusions

Face recognition related technologies are used in many practical applications in our life. With the rapid development of computer vision and artificial intelligence in recent years, face recognition has gone out of laboratory to many practical applications. As a special kind of biological feature and important data source for face recognition, face has many appealing characteristics as follows.

- Firstly, face is a natural and unique feature for each person. This feature, face, is different from each other between different people. This property makes it safe to be used in verification related applications, like real-name verification systems and so on.
- Secondly, it is more convenient to capture the data for face recognition, when compared with many other biological features such as fingerprint, iris, palmprint and so on. We don't have to touch or wear any special device, such as fingerprint scanners or iris collector, to get the face feature. There is also a big range of the distance allowed by face recognition between the person and the face-capture camera, which makes face feature different from other biological features.

Yet, when coming to a face recognition in uncontrolled environments, one big problem is how to weaken the adverse influence induced by noises, such as pixel noise or occlusions, and on the other hand to explore the potential positive effect of some other noises, such as unlabeled data or weakly labeled data. In this thesis, we proposed six different methods to tackle the hard problems of face recognition under uncontrolled environment.

In this thesis, we study the negative effect of deformations and appearance noises, and introduce the concept of directional features into the template orientation and tracking process. Through the extraction of complete information of the image and parallel implementation of GPU, the alignment and tracking accuracy is significantly improved with neglectable computation complexity increase.

To tackle the problem of appearance noise in face images, including Gaussian white noise and missing pixels, directional information and texture information from face images are extracted and a shallow face recognition method is presented. Textural and

directional features are captured at the holistic and part based levels resulting in a robust face descriptor.

To tackle the problem of unsupervised data, we propose a novel multi-view predictive latent space learning (MVP) model and apply it to multi-view clustering.

In case of other variations in face images, such as blur, variations and low resolution. We present how to use high-order representation of features for robust face recognition. A query sample is first modelled as a global Gaussian and then represented jointly on the dictionary of mean vectors and the dictionary of covariance matrices.

To tackle the problem of label noise in datasets, we propose a robust deep face recognition (RDFR) method through an automatic outlier removal. The noisy faces are automatically recognized and removed, which can boost the performance of the learned deep models.

To tackle the problem of insufficient dataset, we explore the possibility of transferring tasks on large-scale datasets to tasks on a limited-scale dataset, and a multi-task deep face recognition method is presented.

Experimental results verify that the methods proposed in this thesis alleviate the noise problems that existed in the uncontrolled real-world environments.

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12):2037–2041, 2006.

[2] L. Wolf, T. Hassner, and Y. Taigman, “Descriptor based methods in the wild”. In Faces in Real-Life Images Workshop in European Conference on Computer Vision, volume 6, 2008.

[3] P. Li, Y. Fu, U. Mohammed, J. Elder, and S. Prince, “Probabilistic models for inference about identity”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(1):144–157, 2012.

[4] S. U. Hussain, T. Napol´eon, F. Jurie, et al, “Face recognition using local quantized patterns”. In British Machine Vision Conference, 2012.

[5] Z. Cao, Q. Yin, X. Tang, and J. Sun, “Face recognition with learning-based descriptor”. In Computer Vision and Pattern Recognition, pages 2707–2714, 2010.

[6] J. Lu, V. E. Liong, G. Wang and P. Moulin, “Joint Feature Learning for Face Recognition”, in IEEE Transactions on Information Forensics and Security, vol. 10, no. 7, pp. 1371-1383, July 2015.

[7] Yandong Wen, Kaipeng Zhang, Zhifeng Li, Yu Qiao. “A Discriminative Feature Learning Approach for Deep Face Recognition”, European Conference on Computer Vision. Springer International Publishing, 2016: 499-515.

[8] Yi Sun, Xiaogang Wang, Xiaoou Tang, “Deep Learning Face Representation from Predicting 10,000 Classes”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1891-1898

[9] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection”. IEEE Transactions on pattern analysis and machine intelligence, 19(7):711–720, 1997.

[10] Dong Yi, Zhen Lei, Shengcai Liao and Stan Z. Li, “Learning Face Representation from Scratch”. arXiv preprint arXiv:1411.7923. 2014.

[11] D. Greene and P. Cunningham, “Producing accurate interpretable clusters from high-dimensional data”, in European Conference on Principles of Data Mining and Knowledge Discovery. Springer, 2005, pp. 486–494.

[12] S. C. H. Hoi, W. Liu, M. R. Lyu, and W. Y. Ma, “Learning distance metrics with contextual constraints for image retrieval”, in CVPR, 2006, pp. 2072–2078

[13] A. Martinez and R. Benavente, “The AR face database”, CVC, Tech. Rep 24, 1998.

[14] Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J., "MS-Celeb-1M: A dataset and benchmark for large scale face recognition". In: European Conference on Computer Vision. Springer (2016)

[15] Chinese Celebrity Face Dataset collected by VIPL. http://vipl.ict.ac.cn/

[16] Zhiwu Huang, Shiguang Shan, Ruiping Wang, Haihong Zhang, Shihong Lao, Alifu Kuerban, Xilin Chen, "A Benchmark and Comparative Study of Video-based Face Recognition on COX Face Database". IEEE Transactions on Image Processing (TIP), 2015.

[17] Ishikawa, T., Matthews, I., Baker, S., "Efficient image alignment with outlier rejection". Technical Report CMU-RI-TR-02-27, Carnegie Mellon University Robotic Institute, 2002.

[18] B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision". In Proceedings of the International Joint Conference on Artificial Intelligence, pages 674–679, 1981.

[19] Hager, G.D., Belhumeur, P.N., "Efficient region tracking with parametric models of geometry and illumination". IEEE Trans. Pattern Anal. Machine Intell. 20 (10), 1025–1039, 1998.

[20] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. "Lucas-Kanade 20 years on: A unifying framework: Part 1". Technical Report CMU-RI-TR-03-01, Carnegie Mellon University Robotics Institute, 2003.

[21] Nicholas Dowson and Richard Bowden, "Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 30, no. 1, January, 2008.

[22] JY Bouguet, "Pyramid Implementation of Lucas Kanade Feature Tracker: Description of the algorithm".

[23] JY Bouguet, "Pyramid Implementation of the Affine Lucas Kanade Feature Tracker: Description of the algorithm".

[24] Katkovnik, V., K. Egiazarian, and J. Astola, "Local Approximation Techniques in Signal and Image Processing", SPIE Press, Monograph Vol. PM157, September 2006.

[25] Katkovnik, V., Egiazarian, K., Astola, J., "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule", J. of Math. Imaging and Vision. 16, 223-235 (2002).

[26] Katkovnik, V., Foi, A., Egiazarian, K., Astola, J., "Directional Varying Scale Approximation for Anisotropic Signal Processing", EUSIPCO Proc. of XII European Signal Process. Conf., 101-104 (2004).

[27] Katkovnik, V., Egiazarian K., Astola J., "A Spatially Adaptive Nonparametric

Regression Image Deblurring", IEEE Trans. Image Process. 14(10), 1469-1478 (2005).

[28] Katkovnik, V., Paliy, D., Egiazarian, K., Astola, J., "Frequency domain blind deconvolution in multiframe imaging using anisotropic spatially-adaptive denoising", Proc. 14th European Signal Process. Conf., (2006).

[29] Katkovnik, V., Astola,J., Egiazarian, K., "Phase local approximation (PhaseLa) technique for phase unwrap from noisy data", IEEE Trans. Image Process., 17, 833-846(2008).

[30] Paliy, D., Katkovnik, V., Bilcu, R., Alenius, S., Egiazarian, K., "Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data", International Journal of Imaging Systems and Technology (IJISP), 17, 105-122(2007).

[31] JY Bouguet, "Pyramid Implementation of the Affine Lucas Knade Feature Tracker: Description of the algorithm".

[32] Foi, A., "Anisotropic nonparametric image processing: theory, algorithms and applications", Ph.D. Thesis. Dip. di Matematica Politecnico di Milano, (2005).

[33] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in SDM, 2008, pp. 822–833

[34] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," TSMC-B, vol. 40, no. 6, pp. 1438–1446, 2010.

[35] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, and J. Jiang, "Sparse unsupervised dimensionality reduction for multiple view data," TCSVT, vol. 22, no. 10, pp. 1485–1496, 2012.

[36] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in NIPS, 2011, pp. 1413–1421.

[37] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in CVPR. IEEE, 2011, pp. 1977–1984.

[38] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph." in AAAI, 2015, pp. 2750–2756.

[39] F. Nie, J. Li, X. Li et al., "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification." IJCAI, 2016.

[40] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity." in ICML, 2013, pp. 352–360.

[41] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in SDM. SIAM, 2013, pp. 252–260.

[42] I. Jolliffe, Principal component analysis. Wiley Online Library, 2002.

[43] B. Thompson, "Canonical correlation analysis," Encyclopedia of statistics in behavioral science, 2005.

[44] S. Akaho, "A kernel method for canonical correlation analysis," arXiv preprint cs/0609071, 2006.

[45] A. Sharma and D. W. Jacobs, "Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch," in CVPR. IEEE, 2011, pp. 593–600.

[46] H. Wang, L. Feng, L. Yu, and J. Zhang, "Multi-view sparsity preserving projection for dimension reduction," Neurocomputing, vol. 216, pp. 286– 295, 2016.

[47] L. Xie, D. Tao, and H. Wei, "Multi-view exclusive unsupervised dimension reduction for video-based facial expression recognition," in IJCAI, 2016.

[48] C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao, "Flexible multi-view dimensionality co-reduction," TIP, vol. 26, no. 2, pp. 648–659, 2017.

[49] N. Chen, J. Zhu, and E. P. Xing, "Predictive subspace learning for multi-view data: a large margin approach," in NIPS, 2010, pp. 361–369.

[50] Z. Lin, G. Ding, M. Hu, and J. Wang, "Multi-label classification via feature-aware implicit label space encoding." in ICML, 2014, pp. 325– 333.

[51] A. Y. Ng, M. I. Jordan, Y. Weiss et al., "On spectral clustering: Analysis and an algorithm," in NIPS, vol. 14, no. 2, 2001, pp. 849–856.

[52] "A matrix factorization approach for integrating multiple data views," in ECML-PKDD. Springer, 2009, pp. 423–438.

[53] W. C. Zhang, S. G. Shan, W. Gao, and H. M. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition," Proc. IEEE Int. Conf. Computer Vision, pp. 786–791(2005).

[54] Z. Lei, S. Liao, R. He, M. Pietikäinen, and S. Z. Li, "Gabor volume based local binary pattern for face representation and recognition," Proc. IEEE Int. Conf. Automatic Face & Gesture Recognition, pp. 1 – 6, (2008).

[55] Z. Wenchao, S. Shiguang, C. Xilin, G.Wen, "Local Gabor Binary Patterns Based on Kullback–Leibler Divergence for Partially Occluded Face Recognition," Signal Processing Letters, IEEE , vol.14, no.11, pp.875-878, Nov. 2007

[56] B. Zhang, S. Shan, X. Chen, W.Gao, "Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition," Image Processing, IEEE Transactions on , vol.16, no.1, pp.57-68, Jan. 2007

[57] Z. Lei, S. Liao, M. Pietikainen, and Z. S. Liu, "Face Recognition By Exploring Information Jointly in Space, Scale and Orientation." IEEE Transactions on Image Processing, vol 20, pp. 247, (2010).

[58] S. Xie, S. Shan, X. Chen, J. Chen, "Fusing Local Patterns of Gabor Magnitude and

Phase for Face Recognition," IEEE Transactions on Image Processing, vol.19, no.5, pp.1349-1361, May 2010.

[59] M. Turk, A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neurosicence, Vol. 3, No. 1, Win. 1991, pp. 71-86.

[60] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, July 1997, pp. 711-720.

[61] A.M. Martinez, A.C. Kak, "PCA versus LDA," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23,pp. 228 - 233, 2001.

[62] Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, Gwo-Jong Yu, "A new LDA-based face recognition system which can solve the small sample size problem," Pattern Recognition, 33 (10) (2000), pp. 1713–1726

[63] H. Yu and J. Yang, "A Direct LDA Algorithm for High- Dimensional Data-With Application to Face Recognition," Pattern Recognition, vol. 34, no. 10, pp. 2067-2070, 2001.

[64] J. Yang, J.Y. Yang, "Why can LDA be performed in PCA transformed space?" Pattern Recognition, 36 (3) (2003), pp. 563–566.

[65] Gui-Fu Lu, Jian Zou, Yong Wang, "Incremental complete LDA for face recognition," Pattern Recognition, 2510-2521, July 2012.

[66] Samaria and F. Fallside., "Automated face identification using hidden markov models," Proceedings of the International Conference on Advanced Mechatronics, 1993.

[67] L. Wiskott, J.-M., Fellous, N. Kruger, C.D. Von Malsburg, "Face Recognition by Elastic Bunch Graph Matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, July 1997, pp. 775-779.

[68] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, July 2002.

[69] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, pp. 2037–2041, (2006).

[70] T. Xiaoyang, B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions," IEEE Transactions on Image Processing, vol.19, no.6, pp.1635-1650, June 2010

[71] B. Zhang, Y. Gao, S. Zhao; J. Liu, "Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor," IEEE Transactions on Image Processing, vol.19, no.2, pp.533-544, Feb. 2010

[72] A. Pentland, B.Moghaddam, T. Starner, "View-based and modular eigenspaces for face recognition," Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[73] V. Blanz, and T. Vetter, "A Morphable model for the synthesis of 3D faces," In Proceedings, SIGGRAPH'99, 187–194,1999.

[74] X. Li, W. Hu , Z. Zhang, H Wang, "Heat Kernel Based Local Binary Pattern for Face Representation," IEEE Signal Processing Letters, vol.17, pp. 308, (2009).

[75] Chih-Chung Chang, Chih-Jen Lin, "LIBSVM: a library for support vector machines," 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[76] Y. Ma, S. Lao, E. Takikawa, and M. Kawade, "Discriminant Analysis in Correlation Similarity Measure Space," Proc. 24th Int'l Conf. Machine Learning, vol. 227, pp. 577-584, 2007.

[77] Y. Fu, S. Yan, T.S. Huang, "Correlation Metric for Generalized Feature Extraction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.30, no.12, pp.2229-2235, Dec. 2008

[78] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., "Caffe: Convolutional architecture for fast feature embedding," In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014)

[79] T. Kobayashi, "Dirichlet-based histogram feature transform for image classification," in Computer Vision and Pattern Recognition, 2014, pp. 3278–3285.

[80] P. Li, X. Lu, and Q. Wang, "From dictionary of visual words to subspaces: Locality-constrained affine subspace coding," in Computer Vision and Pattern Recognition, 2015, pp. 2348–2357.

[81] L. S. Davis, "Covariance discriminative learning: A natural and efficient approach to image set classification," in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2496–2503.

[82] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in International Joint Conference on Artificial Intelligence, 2013, pp. 639–44.

[83] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in European Conference on Computer Vision, 2006, pp. 589–600.

[84] J. Nchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," International Journal of Computer Vision, vol. 105, no. 3, pp. 222–245, 2013

[85] P. Li, H. Zeng, Q. Wang, S. C. Shiu, and L. Zhang, “High-order local pooling and encoding gaussians over a dictionary of gaussians,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3372–3384, 2017.

[86] G. Serra, C. Grana, M. Manfredi, and R. Cucchiara, “Gold: Gaussians of local descriptors for image representation,” Computer Vision and Image Understanding, vol. 134, pp. 22–32, 2015.

[87] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, “Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets,” in CVPR, 2015, pp. 2048–2057.

[88] A. Cherian and S. Sra, “Riemannian dictionary learning and sparse coding for positive definite matrices,” IEEE transactions on neural networks and learning systems, 2017.

[89] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 2, pp. 210–227, 2009.

[90] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 1794–1801.

[91] X. Mei and H. Ling, “Robust visual tracking and vehicle classification via sparse representation,” IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 11, pp. 2259–2272, 2011.

[92] T. Guha and R. K. Ward, “Learning sparse representations for human action recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 8, pp. 1576–1588, 2012.

[93] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, “Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach,” in Computer Vision–ECCV 2012. Springer, 2012, 216–229.

[94] P. Li, Q. Wang, W. Zuo, and L. Zhang, “Log-euclidean kernels for sparse representation and dictionary learning,” in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1601– 1608.

[95] A. Cherian and S. Sra, “Riemannian sparse coding for positive definite matrices,” in European conference on computer vision. Springer, 2014, 299–314.

[96] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, “Tensor sparse coding for positive definite matrices,” IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 3, pp. 592–605, 2014.

[97] S. Zhang, S. Kasiviswanathan, P. C. Yuen, and M. Harandi, "Online dictionary learning on symmetric positive definite manifolds with vision applications." in AAAI, 2015, pp. 3165–3173.

[98] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on the riemannian manifold of symmetric positive definite matrices," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 73–80.

[99] M. Harandi, C. Sanderson, C. Shen, and B. C. Lovell, "Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3120–3127.

[100] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson, "Extrinsic methods for coding and dictionary learning on grassmann manifolds," International Journal of Computer Vision, vol. 114, no. 2-3, pp. 113– 136, 2015.

[101] M. T. Harandi, R. Hartley, B. Lovell, and C. Sanderson, "Sparse coding on symmetric positive definite manifolds using bregman divergences," IEEE transactions on neural networks and learning systems, vol. 27, no. 6, pp. 1294–1306, 2016.

[102] Q. Wang, P. Li, W. Zuo, and L. Zhang, "Raid-g: Robust estimation of approximate infinite dimensional gaussian with application to material recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4433–4441.

[103] Q. Wang, P. Li, and L. Zhang, "G2denet: Global gaussian distribution embedding network and its application to visual recognition," in CVPR, 2017, pp. 2730–2739.

[104] U. Niesen, D. Shah, and G. W. Wornell, "Adaptive alternating minimization algorithms," IEEE Transactions on Information Theory, vol. 55, no. 3, pp. 1423–1429, 2009.

[105] L. Sharan, R. Rosenholtz, and E. H. Adelson, "Material perception: What can you see in a brief glance?" Journal of Vision, vol. 9, no. 8, 784–784, 2009.

[106] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in CVPR, 2008, 1–8.

[107] Z. Liao, J. Rock, Y. Wang, and D. Forsyth, "Non-parametric filtering for geometric detail extraction and material representation," in CVPR, June 2013.

[108] R. Wang, H. Guo, L. S. Davis, and Q. Dai, "Covariance discriminative learning: A natural and efficient approach to image set classification," in CVPR, 2012, pp. 2496–2503.

[109] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification." in ICML, 2015, pp. 720–729.

[110] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[111] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on back-ground samples," in Asian Conference on Computer Vision. Springer, 2009, pp. 88–97.

[112] R. Wang, S. Shan, X. Chen, and W. Gao, "Manifold-manifold distance with application to face recognition based on image set," in CVPR, 2008, pp. 1–8.

[113] R. Wang and X. Chen, "Manifold discriminant analysis," in CVPR, 2009, 429–436.

[114] H. Cevikalp and B. Triggs, "Face recognition based on image sets," in CVPR, 2010, pp. 2567–2573.

[115] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices," in European conference on computer vision. Springer, 2014, pp. 17–32.

[116] LeCun, Y., Bengio, Y., Hinton, G. "Deep learning," Nature 521(7553), 436–444 (2015)

[117] Schroff, F., Kalenichenko, D., Philbin, J., "Facenet: A unified embedding for face recognition and clustering," In: Computer Vision and Pattern Recognition (CVPR). pp. 815–823 (2015)

[118] Sun, Y., Liang, D., Wang, X., Tang, X., "Deepid3: Face recognition with very deep neural networks," arXiv preprint arXiv:1502.00873 (2015)

[119] Ariz, M., Bengoechea, J.J., Villanueva, A., Cabeza, R., "A novel 2d/3d database with automatic face annotation for head tracking and pose estimation," Computer Vision and Image Understanding 148, 201–210 (2016)

[120] Manwani, N., Sastry, P.S., "Noise tolerance under risk minimization," IEEE Transactions on Cybernetics 43(3), 1146 (2011)

[121] Patrini, G., Nielsen, F., Nock, R., Carioni, M., "Loss factorization, weakly supervised learning and label noise robustness," In: International Conference on International Conference on Machine Learning. pp. 708–717 (2016)

[122] Gao, W., Wang, L., Li, Y.F., Zhou, Z.H., "Risk minimization in the presence of label noise," In: AAAI. pp. 1575–1581 (2016)

[123] Zhang, J., Sheng, V.S., Li, T., Wu, X., "Improving crowdsourced label quality using noise correction," IEEE Transactions on Neural Networks and Learning

Systems (2017)

[124] Brodley, C.E., Friedl, M.A., "Identifying and eliminating mislabeled training instances," In: Thirteenth National Conference on Artificial Intelligence. pp. 799–805 (1996)

[125] Schroff, F., Criminisi, A., Zisserman, A., "Harvesting image databases from the web," IEEE transactions on pattern analysis and machine intelligence 33(4), 754–766 (2011)

[126] Li, L.J., Fei-Fei, L., "OPTIMOL: automatic online picture collection via incremental model learning," International journal of computer vision 88(2), 147–168 (2010)

[127] Collins, B., Deng, J., Li, K., Fei-Fei, L., "Towards scalable dataset construction: An active learning approach," Computer Vision–ECCV 2008 pp. 86–98 (2008)

[128] Kim, J., Scott, C.D., "Robust kernel density estimation," Journal of Machine Learning Research 13(Sep), 2529–2565 (2012)

[129] Elhamifar, E., Sapiro, G., Vidal, R., "See all by looking at a few: Sparse modeling for finding representative objects," In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 1600–1607. IEEE (2012)

[130] Liu, W., Hua, G., Smith, J.R., "Unsupervised one-class learning for automatic outlier removal," In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3826–3833 (2014)

[131] He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385 (2015)

[132] Liu X, Kan M, Wanglong W U, et al. "VIPLFaceNet: an open source deep face recognition SDK," Frontiers of Computer Science, 2017, 11(2):208-218.

[133] Keys, R., "Cubic Convolution Interpolation for Digital Image Processing", IEEE Trans on ASSP, vol ASSP-29, No. 6, Dec 1981

[134] V. N. Vapnik, "Statistical Learning Theory," New York: Wiley, 1998.

[135] A. Georghiades, P. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23,no. 6, pp. 643-660, June 2001.

[136] K. Lee, J. Ho, and D. Kriegman, "Acquiring Linear subspaces for face recognition under variable lighting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 5, pp. 684–698, 2005.

[137] T. Sim, S. Baker, M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) database, Automatic Face and Gesture Recognition," 2002. Proceedings. Fifth IEEE International Conference on , vol., no., pp.46-51, 20-21 May 2002.

[138] P. J. Phillips, H. Wechsler, J. Huang, P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," Image Vis. Comput. J., vol. 16, no. 5, pp. 295–306, 1998.

[139] P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[140] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[141] Yuan J, Egiazarian K, "Anisotropic multi-scale Lucas-Kanade pyramid," Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V. 2011, 7881(1):78810O-78810O-12.

[142] Mehta R, Yuan J, Egiazarian K, "Face recognition using scale-adaptive directional and textural features," Pattern Recognition, 2014, 47(5): 1846-1858.

[143] Jirui Yuan, Ke Gao, Pengfei Zhu, and Karen Egiazarian, "Multi-view Predictive Latent Space Learning," Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.06.022

[144] Jirui Yuan, Hao Cheng, and Karen Egiazarian. "Co-regularized Sparse Representation of Gaussians for Pattern Classification," In submission to Pattern Recognition.

[145] Yuan J, Ma W, Zhu P., and Egiazarian K, "Robust Deep Face Recognition with Label Noise," International Conference on Neural Information Processing. Springer, Cham, 2017: 593-602.

[146] Yuan J, Ma W, Zhu P, and Egiazarian K. "Multi-task Deep Face Recognition," Chinese Conference on Biometric Recognition. Springer, Cham, 2017: 183-190.

Publications

Publication I: Anisotropic multi-scale Lucas-Kanade pyramid. Yuan J, Egiazarian K. Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V. 2011, 7881(1): 78810O-78810O-12.

Anisotropic multi-scale Lucas-Kanade pyramid

Jirui Yuan, Karen Egiazarian

Jirui Yuan, Karen Egiazarian, "Anisotropic multi-scale Lucas-Kanade pyramid," Proc. SPIE 7881, Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V, 78810O (11 February 2011); doi: 10.1117/12.878819

Event: IS&T/SPIE Electronic Imaging, 2011, San Francisco Airport, California, United States

Anisotropic Multi-Scale Lucas-Kanade Pyramid

Jirui Yuan*[a], Karen Egiazarian*[a]
[a]Department of Signal Processing, Tampere University of Technology, Tampere, Finland, FI-33720

ABSTRACT

The Lucas-Kanade (LK) algorithm provides a smart iterative parameter-update rule for efficient image alignment, and it has become one of the most widely used techniques in computer vision. Applications range from optical flow and tracking to layered motion, mosaic construction, and face coding. In this paper, we propose a novel Anisotropic Multi-Scale Lucas-Kanade Pyramid (AMSLKP) method. By extracting image pyramids from the original images and iteratively implementing LK algorithm at each level, the Lucas-Kanade Pyramid (LKP) gained better robustness and accuracy. Moreover, instead of calculating gradients in single direction with fixed scale sizes, this paper introduces anisotropic local polynomial approximation (LPA) and intersection of conference intervals (ICI) method to the LKP. The proposed AMSLKP method first calculates the directional estimates and gradients with multiple scales; then for each direction, it adaptively selects the optimum scale for each pixel in the image using ICI rule; at last, the estimate and gradients of the distorted image are computed by fusing the directional results together. The proposed method is evaluated in different noise conditions with various distortion levels. Experimental results show that the AMSLKP method improves the accuracy by more than forty percent compared to LKP method.

Keywords: Local Polynomial, Lucas-Kanade Pyramid, ICI Rule, Adaptive Multi-Scale

1. INTRODUCTION

The alignment or registration of a pair of images is an operation required in many applications such as image mosaicking, simultaneous localization and tracking, and multimodal image alignment. Among these applications, the most remarkable one is the Active Appearance Model[3], which is also a key part of face recognition and hallucination. AAM generates faces with varying expression and pose by modifying its appearance image and warping it into different shapes. The key of AAM is the efficient image alignment which first defines a geometric deformation scheme and then warps one image onto another such that they become as similar as possible to some criterion. To efficiently align a template image to a reference image, Lucas and Kanade made a series of pioneer work by minimizing the sum of the squared difference similarity function. The earliest image alignment algorithm was the Lucas-Kanade algorithm[1]. In this algorithm, iterative parameter updates to alignment parameters are obtained by multiplying the Jacobian with the inverse Hessian of the similarity function. However, these LK methods only provide limited robustness to noise on image and template. There are also some methods that focused on proposing new similarity functions. Instead of sum of squared differences used in the so-called Lucas-Kanade framework[2,4] utilized Mutual Information (MI) which tolerates nonlinear relationships between the intensities in image and is robust to noise, and developed an inverse compositional formulation for MI. Then in 2009, the concept of Lucas-Kanade pyramid[6] was proposed focusing on the application of optical flow. Later on, the concept of pyramid is extended to affine Lucas-Kanade feature tracking method[7]. By extracting image pyramids from the original images and iteratively implementing LK algorithm for each level image, the LKP method gained a lot in robustness and accuracy. Yet there are still some aspects that can be considered which were ignored in all above mentioned methods. First, one important feature of the LK methods is that the parameters are iteratively updated until the variation in parameters or function values becomes sufficiently small, and the image gradients are utilized frequently in the iterative parameter update process. For noisy images, the incorrect calculation of the gradients may result in severe noise amplification and propagation through the LK iterations resulting in intense degradation in the accuracy of parameter estimations. In this paper, we apply a powerful tool, local polynomial approximation (LPA)[5] into the LKP method and propose a novel Anisotropic Multi-Scale Lucas-Kanade Pyramid (AMSLKP) method which for the first time introduces the concept of multiple directions to LK feature tracking process. The LPA method has been utilized in a number of applications related to image processing such as image denoising[8], image deblurring[9], image

Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V, edited by David Akopian, Reiner Creutzburg, Cees G. M. Snoek, Nicu Sebe, Lyndon Kennedy, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7881, 78810O · © 2011 SPIE-IS&T · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.878819

reconstruction[12], phase unwrapping[11], and color filter array interpolation[13]. Instead of calculating gradients in only one direction with a fixed scale, we utilize multiple directions with multiple scales and adaptively select the optimum scale for each pixel. Besides, the estimate of the noisy image is also calculated and then utilized in the LK pyramid.

The rest of this paper is arranged as follows: After an introduction to LK affine feature tracking methods in Section 2, Anisotropic Multi-Scale LK Pyramid (AMSLKP) method is proposed in Section 3. In Section 4, the experimental results of performance of the proposed algorithm in different noise conditions are given. Finally, Section 5 draws the conclusions.

2. BACKGROUND: LUCAS-KANADE AFFINE FEATURE TRACKER

Let I and T be two grayscale images. The two quantities $I(x) = I(x_0, x_1)$ and $T(x) = T(x_0, x_1)$ are then the gray-scale values of these two images at the location $x = [x_0 \ x_1]^T$, where x_0 and x_1 are the two pixel coordinates of a generic image point x. Let $W(x;p)$ denote the parameterized set of affine warp, where $p = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6]^T$ is a vector of parameters. Herein,

$$W(x;p) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ 1 \end{pmatrix} \tag{1}$$

For LK affine feature tracking, the goal is to find the 6 parameters $p = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6]^T$ such that $T(x)$ and $I\big(W(x;p)\big)$ are 'similar', which is to minimize the sum of squared errors between the template T and image I warped back onto the coordinate frame of the template:

$$\epsilon(p_1, p_2, p_3, p_4, p_5, p_6) = \sum_x \Big(T(x) - I\big(W(x;p)\big)\Big)^2 \tag{2}$$

To optimize the expression in Equation (2), the Lucas-Kanade algorithm assumes that a current estimate of p is known and then iteratively solves for increments to the parameters Δp; i.e. the following expression is minimized[2]:

$$\sum_x \Big(T(x) - I\big(W(x;p+\Delta p)\big)\Big)^2 \tag{3}$$

with respect to Δp, and then the parameters are updated:

$$p \leftarrow p + \Delta p \tag{4}$$

To the Equation (4), we apply a first order Taylor expansion on $I\big(W(x;p+\Delta p)\big)$ to give:

$$\sum_x \left[I\big(W(x;p)\big) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x)\right]^2 \tag{5}$$

where ∇I is the gradient of the image I evaluated at $W(x;p)$. A partial derivative with respect to Δp is then obtained:

$$2\sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \left[I\big(W(x;p)\big) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x)\right] \tag{6}$$

Assuming a locally parabolic shape and setting the partial derivative to zero gives a closed form solution for updating p, which minimizes Equation (5):

$$\Delta p = H^{-1} \sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \big[T(x) - I\big(W(x;p)\big)\big], \tag{7}$$

$$H = \sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \left[\nabla I \frac{\partial W}{\partial p}\right] \tag{8}$$

The warp parameter must be iteratively computed and updated until the variation in parameters or function values becomes sufficiently small.

3. ANISOTROPIC MULTI-SCALE LUCAS-KANADE PYRAMID METHOD

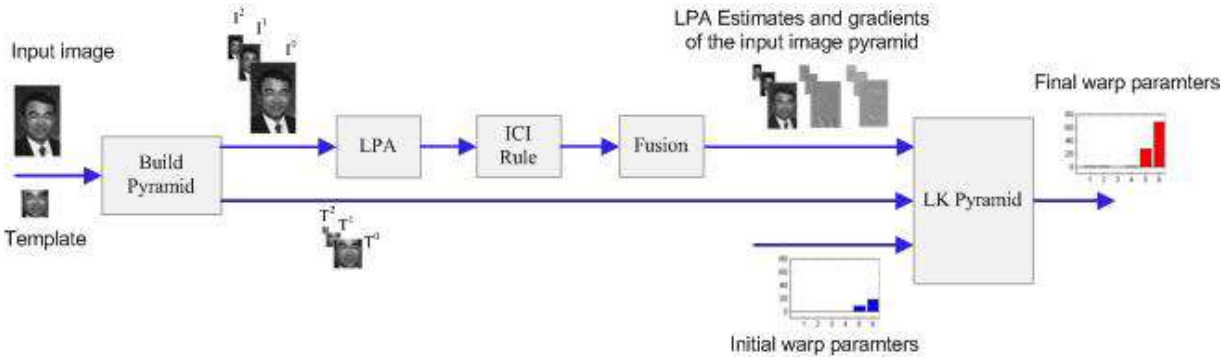


Figure 1. Diagram for the proposed Adaptive Multi-Scale Lucas-Kanade Pyramid

The proposed AMSLKP framework is shown in Figure 1. The parameter estimation process consists of the following steps: 1) generate the image $\{I^L\}_{L=0,1,\dots L_m}$ and template $\{T^L\}_{L=0,1,\dots L_m}$ pyramids; 2) generate the local polynomial kernels in a number of directions and several scales, calculate the estimates and gradients of the images using the generated kernels; 3) utilize ICI rule to select the point-wise adaptive scale and the optimum estimates and gradients for all directions; 4) fuse the estimates and gradients from all directions together to yield the estimate and gradient $\{\hat{I}^L, \nabla\hat{I}^L\}_{L=0,1,\dots L_m}$; 5) evaluate the warp parameters using LKP method. All these steps are discussed in detail in the following sections.

3.1 Building Image Pyramids

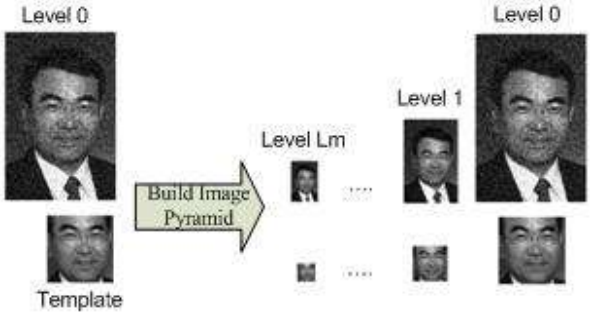


Figure 2. Process of building pyramidal images

Define the pyramidal representation of a generic image I of size $n_x \times n_y$. Let $I^0 = I$ be the 0^{th} level image. Image I^0 is the highest resolution image (raw image). The image size at 0^{th} level is defined as $n_x^0 = n_x,\ n_y^0 = n_y$. The pyramidal representation is then built in a recursive fashion: compute I^1 from I^0, then I^2 from I^1, and so on… Let $L = 1,2, \dots$ be a generic pyramidal level, and let I^{L-1} be the image at level $L-1$. Denote n_x^{L-1}, n_y^{L-1} the width and height of I^{L-1}. Then the pyramidal images $\{I^L\}_{L=0,1,\dots L_m}$ and $\{T^L\}_{L=0,1,\dots L_m}$ are constructed recursively[7] using (9).

$$
\begin{aligned}
I^L(x,y) = \ & \frac{1}{4} I^{L-1}(2x, 2y) \\
& + \frac{1}{8}\left(I^{L-1}(2x-1,2y) + I^{L-1}(2x-1,2y) + I^{L-1}(2x+1,2y) + I^{L-1}(2x,2y+1)\right) \\
& + \frac{1}{16}\left(I^{L-1}(2x-1,2y-1) + I^{L-1}(2x+1,2y+1) + I^{L-1}(2x+1,2y-1)\right. \\
& \left. + I^{L-1}(2x-1,2y+1)\right)
\end{aligned} \tag{9}
$$

The width n_x^L and height n_y^L of I^L are the largest integers that satisfy the conditions:

$$n_x^L \leq \frac{n_x^{L-1}+1}{2}, \quad n_y^L \leq \frac{n_y^{L-1}+1}{2} \tag{10}$$

Using equations (9) and (10), the image $\{I^L\}_{L=0,1,\dots L_m}$ and template $\{T^L\}_{L=0,1,\dots L_m}$ pyramids are recursively constructed from image I and template T, as shown in Fig.2.

3.2 Local Polynomial Approximation

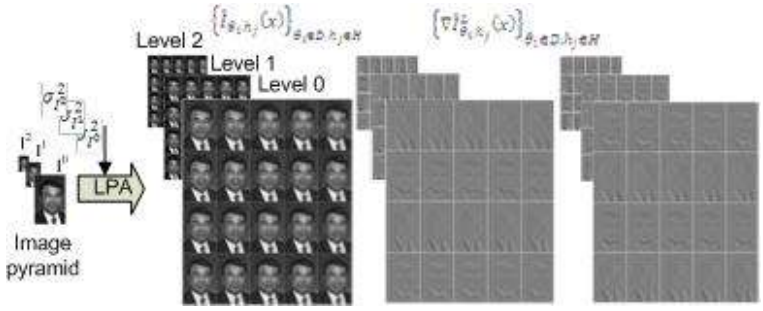


Figure 3. Taking 3-level pyramid, utilizing the 0th and 1st order local polynomial kernels with 4 directions and 5 scales, the figure shows the representation of LPA estimations and gradients

For each level image I^L, we define $\{\theta_i \in D\}$ as the angle between the i^{th} direction and horizontal line, and $h_j \in H = \{h_1, h_2, \dots, h_J\}$ as the scale set of local polynomial kernels. For a given direction θ_i and a given scale h_j, the corresponding local polynomial kernels $g_{\theta_i,h_j}(x,X_s)|_{\theta_i \in D, h_j \in H}$, are generated[5] to approximate the estimate $\hat{I}^L_{\theta_i,h_j}$ from the L^{th} level image I^L with the corresponding variance $\sigma^2_{\hat{I}^L_{\theta_i,h_j}}$:

$$\hat{I}^L_{\theta_i,h_j}(x) = \sum_s g_{\theta_i,h_j}(x,X_s) I^L(X_s) \tag{11}$$

$$g_{\theta_i,h_j}(x,X_s) = w_{\theta_i,h_j}(x-X_s)\phi_h^T(x-X_s)\psi_{h_j}^{-1}\phi_h(0) \tag{12}$$

$$\sigma^2_{\hat{I}^L_{\theta_i,h_j}}(x) = (\sigma^L)^2 \sum_s g^2_{\theta_i,h_j}(x,X_s) \tag{13}$$

$$\psi_{h_j} = \sum_s w_{\theta_i,h_j}(x-X_s)\phi_h(x-X_s)\phi_h^T(x-X_s) \tag{14}$$

$$\phi_h(x) = \frac{(-1)^{|k|}x^k}{k!}, \qquad k \in Z^2 \colon \forall |k| \leq m \tag{15}$$

Besides, the first order local polynomial kernels $g^{(r)}_{\theta_i,h_j}(x,X_s)|_{\theta_i \in D, h_j \in H}$ is also generated in order to approximate the derivatives $\nabla \hat{I}^L_{\theta_i,h_j}(x)$:

$$\nabla \hat{I}^L_{\theta_i,h_j}(x) = \sum_s g^{(r)}_{\theta_i,h_j}(x,X_s) I^L(X_s) \tag{16}$$

$$g^{(r)}_{\theta_i,h_j}(x,X_s) = \left(-\frac{1}{h}\right)^{|r|} w_{\theta_i,h_j}(x-X_s)\phi_h^T(x-X_s)\psi_{h_j}^{-1}\phi^{(r)}(0) \tag{17}$$

Here, $w_{\theta_i,h_j}(x,-X_s)$ is a directional Gaussian window in direction θ_i with scale h_j; m is the order of polynomial; while ϕ_h is a vector of polynomial for LPA and the length of the vector is equal to $m+1$.

Figure 3 presents how the local polynomial estimates and gradients are calculated for one particular case when there are three levels in the pyramid, and the 0th and 1st order local polynomial kernels with four directions and five scales are utilized.

3.3 ICI Rule

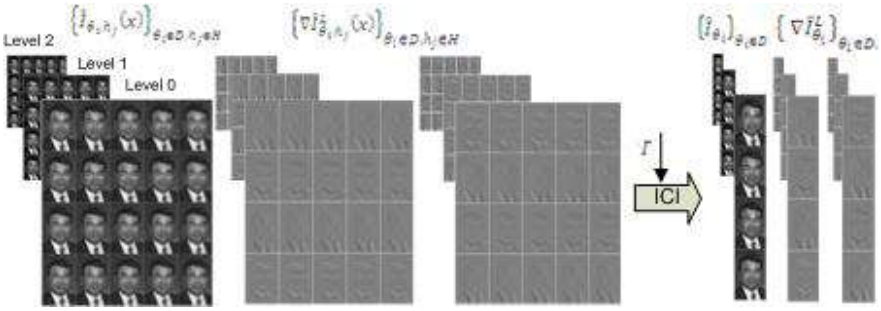


Figure 4. Taking 3-level pyramid, utilizing the 0th and 1st order local polynomial kernels with 4 directions and 5 scales, the figure gives the representation of how ICI rule is utilized to generate the optimum directional estimates and gradients

From Sec 2.2, for a given direction θ_i, we can generate all the estimates and gradients $\left\{\hat{I}^L_{\theta_i,h_j}, \nabla\hat{I}^L_{\theta_i,h_j}\right\}_{h_j \in H}$ for all scales $h_j \in H$. Then ICI rule[5] is utilized in order to find the point-wise optimum scale $h^+ = h_{j_+}$ for all directions $\{\theta_i \in D\}$ and thus the directional estimate $\hat{I}^L_{\theta_i} = \hat{I}^L_{\theta_i,h_+}$, and directional gradients $\nabla\hat{I}^L_{\theta_i} = \nabla\hat{I}^L_{\theta_i,h_+}$. For a scale h_j, the possible intersection at point x is

$$Q_j = \left[L_j, U_j\right],$$

where

$$U_j = \hat{I}^L_{\theta_i,h_j}(x) + \Gamma \cdot \sigma_{\hat{I}^L_{\theta_i,h_j}}(x), \tag{17}$$

$$L_j = \hat{I}^L_{\theta_i,h_j}(x) - \Gamma \cdot \sigma_{\hat{I}^L_{\theta_i,h_j}}(x)$$

Let $\overline{L}_{j+1} = max\{\overline{L}_j, L_{j+1}\}$, $\underline{U}_{j+1} = max\{\underline{U}_j, U_{j+1}\}$. Find the largest j when $\overline{L}_j \le \underline{U}_j|_{j=1,2,\ldots,J}$ is still satisfied. Denote this largest value j_+. Then this j_+ is the largest of those j for which the confidence intervals Q_j have a point in common and the ICI adaptive scale is $h^+ = h_{j_+}$. In such a way, the directional estimates and gradients $\left\{\hat{I}^L_{\theta_i}, \nabla\hat{I}^L_{\theta_i}\right\}_{\theta_i \in D, L=0,1,\ldots L_m}$ for all level images in the pyramid can be calculated using (18). The whole process is shown in Figure 4.

$$\hat{I}^L_{\theta_i}(x) = \hat{I}^L_{h,\theta_i}(x)|_{h\,=\,h_+(x,\,\theta_i)}, \text{and } \nabla\hat{I}^L_{\theta_i}(x) = \nabla\hat{I}^L_{h,\theta_i}(x)|_{h\,=\,h_+(x,\,\theta_i)} \tag{18}$$

3.4 Estimate and Derivative Fusion

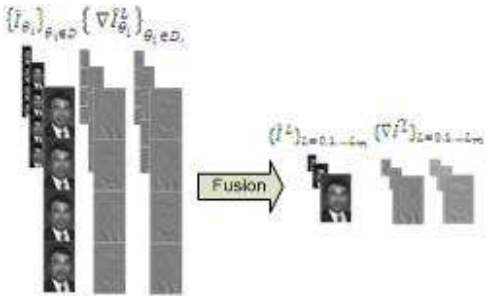


Figure 5. Taking 3-level pyramid, utilizing the 0th and 1st order local polynomial kernels with 4 directions and 5 scales, the figure gives the representation of how directional estimates and gradients are fused to generate the final estimates and gradients $\left\{\hat{I}^L, \nabla\hat{I}^L\right\}_{L=0,1,\ldots L_m}$

After selecting the best adaptive-scale for each direction θ_i using ICI rule, we obtain a set of adaptive estimates and gradients for each pixel x. These estimates and gradients should be fused into a single final estimate and gradients. For the fusion of directional estimates $\left\{\hat{I}^L_{\theta_i}\right\}_{\theta_i \in D, L=0,1,\ldots L_m}$, we utilize the multi-window estimate with the inverse variances as

the weights λ_i for the linear fusion as following:

$$\hat{I}^L(x) = \sum\nolimits_{\theta_i \in D} \lambda_i \hat{I}^L_{\theta_i}(x) \tag{19}$$

$$\lambda_i = \frac{\sigma^{-2}_{\hat{I}^L_{\theta_i}}(x)}{\sum_{\theta_i \in D} \sigma^{-2}_{\hat{I}^L_{\theta_i}}(x)}, \qquad \sigma^2_{\hat{I}^L_{\theta_i}}(x) = \sigma^2_{\hat{I}^L_{\theta_i, h_j}}(x) \; |_{h = h_+(x, \theta_i)} \tag{20}$$

Taking a single level image I from the pyramid, the gradient ∇I can be found by minimizing the weighted mean-square criterion:

$$J = \sum_{i=1}^{K} \frac{1}{\sigma^2_{\theta_i}} \left[\hat{I}^{(1,0)}_{\theta_i} - \left(\partial_{x_0} \hat{I}_{\theta_i} \cdot \cos\theta_i + \partial_{x_1} \hat{I}_{\theta_i} \cdot \sin\theta_i \right) \right]^2 \tag{21}$$

where the weights $\sigma^2_{\theta_i}$ are the variances of the estimates $\hat{I}^{(1,0)}_{h,}$. In the vector-matrix notation, J can be rewritten as:

$$J = \left(\hat{I}^{(1,0)}_h - B\nabla I \right)^T \Lambda \left(\hat{I}^{(1,0)}_h - B\nabla I \right) \tag{22}$$

where $\hat{I}^{(1,0)}_\theta = \left(\hat{I}^{(1,0)}_{\theta_1}, \ldots, \hat{I}^{(1,0)}_{\theta_K} \right)^T$ is a vector of the estimates, $\Lambda = diag\left\{ 1/\sigma^2_{\theta_1}, \ldots, 1/\sigma^2_{\theta_K} \right\}$ is a diagonal matrix, and $B = (B_i)_{K\times 2}, B_i = (\cos\theta_i, \sin\theta_i)$.By minimizing the mean-square error, the estimate of the gradient can be calculated by:

$$\nabla \hat{I} = (B^T \Lambda \mathrm{B})^{-1} B^T \Lambda \hat{I}^{(1,0)}_\theta \tag{23}$$

Utilizing equations (19) and (22) for directional estimates $\hat{I}^L_{\theta_i}$ and gradients $\nabla \hat{I}^L_{\theta_i}(x)$ for all the levels in the pyramid, we can generate the final estimates and gradients $\left\{ \hat{I}^L, \nabla \hat{I}^L \right\}_{L=0,1,\ldots L_m}$. The fusion process is described in Figure 5.

3.5 Lucas-Kanade Pyramid

From pyramidal view, the warp parameter p is also the warp parameter for the 0^{th} level image $\hat{I}^0$ i.e. $p = p^0$, where p^L is evaluated by minimizing the difference between the L^{th} level warped directional estimate $\hat{I}^L\left(W(p^L + \Delta p^L)\right)$ and the template image T, i.e.

$$p^L = \min_{\Delta p^L} \left\{ \sum\nolimits_x \left(\hat{I}^L\left(W(p^L + \Delta p^L)\right) - T^L \right)^2 \right\} \tag{24}$$

Taking LK method (4, 5, and 6) into (18), we have

$$\Delta p^L = \left(\widehat{H}^L \right)^{-1} \sum_x \left[\nabla \hat{I}^L \frac{\partial W}{\partial p} \right]^T \left[T^L(x) - \hat{I}^L\left(W(x; p^L)\right) \right] \tag{25}$$

$$\widehat{H}^L = \sum_x \left[\nabla \hat{I}^L \frac{\partial W}{\partial p} \right]^T \left[\nabla \hat{I}^L \frac{\partial W}{\partial p} \right] \tag{26}$$

p^L is calculated by iteratively updating parameters:

$$p^L \leftarrow p^L + \Delta p^L \tag{27}$$

The parameter update process is recursively implemented from ${L_m}^{\text{th}}$ level down to 0^{th} level by evaluating the initial guess for $(L-1)^{\text{th}}$ level parameter from L^{th} level using:

$$\boldsymbol{p}^{L-1}_{init} = [{p_1}^L \; {p_2}^L \; {p_3}^L \; {p_4}^L \quad {p_5}^L \times 2 \quad {p_6}^L \times 2] \tag{28}$$

4. EXPERIMENTS

4.1 Experiment conditions

Since one application of the proposed LK tracking method is to prepare face data for face hallucination and recognition systems, our proposed method is evaluated over several faces chosen from FERET dataset[14]. In order to give a more general and comprehensive analysis of our proposed method, totally we choose 10 frontal faces from different races and different genders and label them as I. For each face, we manually select the centre part of the face as template T. Then these template faces $\{T\}$ together with the original faces $\{I\}$ are used to LKP[7] and the proposed AMSLKP method. The observation image I and template T may be degraded by the addition of Gaussian white noise with a range of standard deviations. The experiments are conducted over 2 different scenarios depending on whether template T also suffers from noise; here, we label these two scenarios as S1 and S2:

S1. Image I is noisy while template T is not;
S2. Image I and template T are both noisy.

Moreover, for both scenarios, various noise intensity are also considered. To parameterize, the standard deviation σ of the added noise is chosen from the noise condition set $\sigma_I, \sigma_T \in \{0, 4, 8, 12, 16\}$, and the perturbation noise variance σ_s is chosen from set: $\sigma_s \in \{2,4,6,8,10\}$. Since the 6 parameters in the affine warp have different units, we use the following error measure rather than the errors in the parameters. Given the current estimate of the warp, we compute the destination of the 3 canonical points and compare them with the correct locations. We compute the Root Mean Square Error over the 3 points of the distance between their current and correct locations. In order to yield more precise results, 100 tests are implemented for each noise condition, and the overall result for a given noise condition is averaged from the 100 test results.

Formula (17) reveals the role of the threshold parameter Γ in ensuring the fidelity of the adaptive estimate $\hat{I}_{\theta_i,h_+}$since the Γ value will directly influence the choice of adaptive scales. Roughly speaking, ICI selects the coarsest scale estimate that is statistically compatible with all finer scales. This means that adaptively, for each pixel, ICI allows the maximum degree of smoothing stopping before over-smoothing begins[15]. Thus here we illustrate the impact of the threshold parameter Γ on the Root Mean Square Point Error (RMSPE). Also we manipulate the range of $\theta_i \in D$ by tuning number of directions, range of $h_j \in H = \{1, 2, \ldots\ h_{max}\}$ by varying the maximum scale h_{max} and the structure of LPA kernels g_{θ_i,h_j}. In all these tests, 3 levels are utilized in the pyramid generation process, i.e. $L_m = 2$, while in each level, 5 iterations are adopted when estimating parameters.

4.2 Kernel Types

Starting with kernel structures, considering the symmetry of local polynomial kernels g_{θ_i,h_j} in (11), 3 types of kernels are evaluated including asymmetric-line-kernel, asymmetric-kernel and part-symmetric-kernel which means the kernel is symmetric on x0-direction and asymmetric on x1-direction. To simplify, we label these kernels as ASLK, ASK and PSK in the following experiments. Figure 6.a, 6.b and 6.c respectively give the structures of above kernels in the case of 4 directions with scale size equals 2, i.e. $\{g_{\theta_i,h}\}_{\theta_i \in \{0,\frac{\pi}{2},\pi,\frac{3\pi}{2}\},h=2}$.

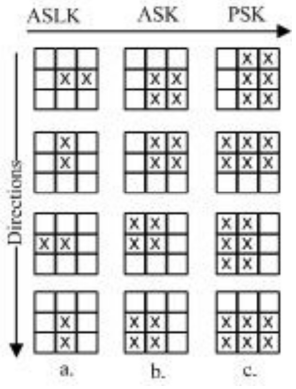


Figure 6. Structure of different kernel types for a particular case

To give a clearer analysis of how different kernel types may influence the accuracy of the proposed AMSLKP method, in all the tests we first fix the scale set $H = \{1, 2, 3, 4\}$ and the direction set $D = \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, and then we evaluate the

AMSLKP method using the ASLK, ASK and PSK kernels and compare the results with LKP[7] methods. The evaluation process is conducted for variance noise conditions with respect to a wide Γ range from 0.2 to 3 with step-size equaling 0.1. Performances results show that ASK kernel provides better accuracy of parameters. In Figure 7, the influence of different kernels together with the effect of threshold parameter Γ is presented for 2 noise conditions: $\sigma_s = 8, \sigma_I = 20, \sigma_T = 0$ and $\sigma_s = 8, \sigma_I = 24, \sigma_T = 0$.

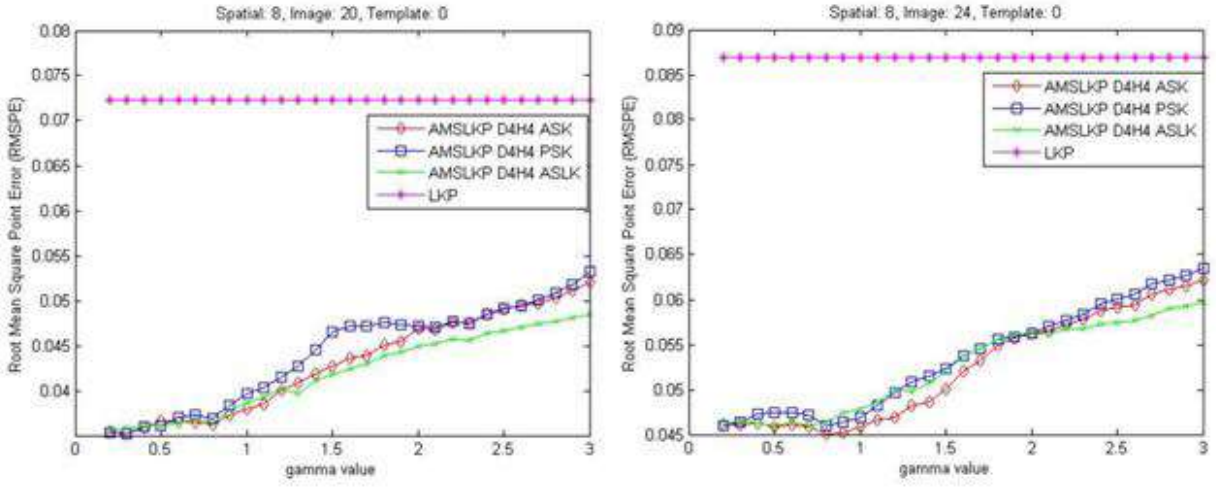


Figure 7. RMSPE- Γ performance comparison between various kernel types

4.3 Number of Directions

Other than kernel types, the sector division of local polynomial kernels may also influence the accuracy performance. So in this part, we fix the scale set $H = \{1, 2, 3, 4\}$ for ASK kernel, and analyze how the number of direction may affect the performance. In the tests, we evaluate the performance when 2, 4, and 8 directions, i.e. $\theta_i \in \{0, \pi\}$, $\left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, and $\left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\right\}$, are utilized in the local polynomial sector division. Figure 8 gives the RMSPE performance over the same Γ range from 0.2 to 3. The results show that best result is provided by dividing the sectors in four directions $\theta_i \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$.

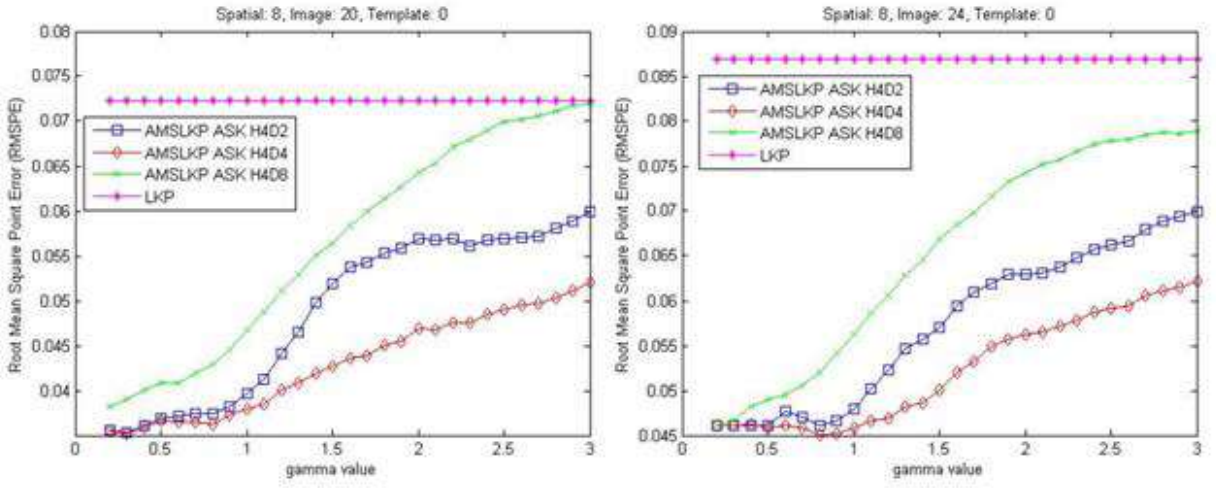


Figure 8. RMSPE- Γ performance comparison between different sector divisions

4.4 Different Scales

The threshold parameter Γ plays a relevant role in the selection of the adaptive scale. From formula (14) we observe that a too large Γ makes the intersection of the confidence intervals more likely to be non-empty[15], and that as a consequence the chosen adaptive scale may be rather large. Analogously, with a small Γ, the empty intersection is likely to happen at the smaller scale. From a purely theoretical point of view, as the Γ value increases, the probability of over-smoothing the image $\hat{I}^L$ increases. Thus the accuracy of proposed method will degrade. So theoretically, if we extend the scale set by tuning the h_{max} value, the AMSLKP method should intensively degrade for large Γ values as h_{max} increases. Figure 9 shows promising results as h_{max} increases from 2 to 8 with step-size equaling 2.

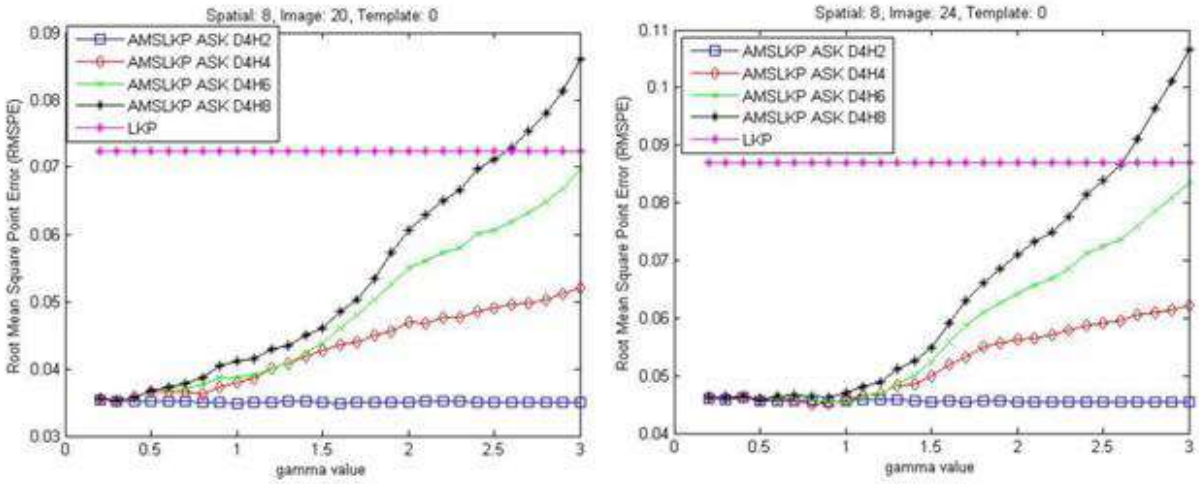


Figure 9. RMSPE- Γ performance comparison between different kernel scales

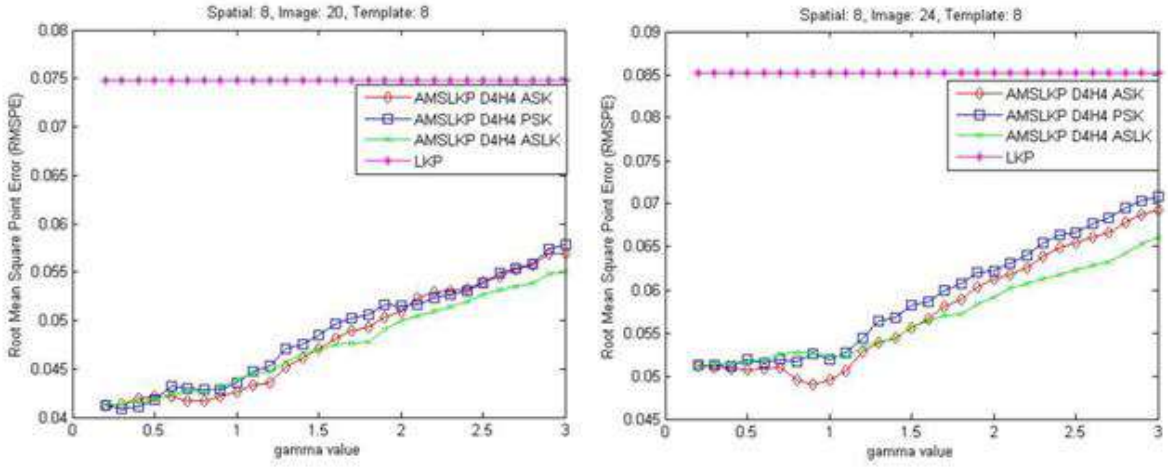
4.5 Results for Scenario 2

All above tests on kernel types, kernel directions and kernel scales are also conducted for the 2nd scenario S2 when both image I and template T are added with white Gaussian noise. Figure 10 gives similar results for 2 typical cases when $\sigma_s = 8, \sigma_I = 16, \sigma_T = 8$ and $\sigma_s = 8, \sigma_I = 24, \sigma_T = 8$. As we can see from the plots, all conclusions derived from Scenario 1 holds. The optimum Γ value could be set to 0.6, besides the maximum scale h_{max} can be set to 4 with the number of directions set to 4 and kernel type ASK.

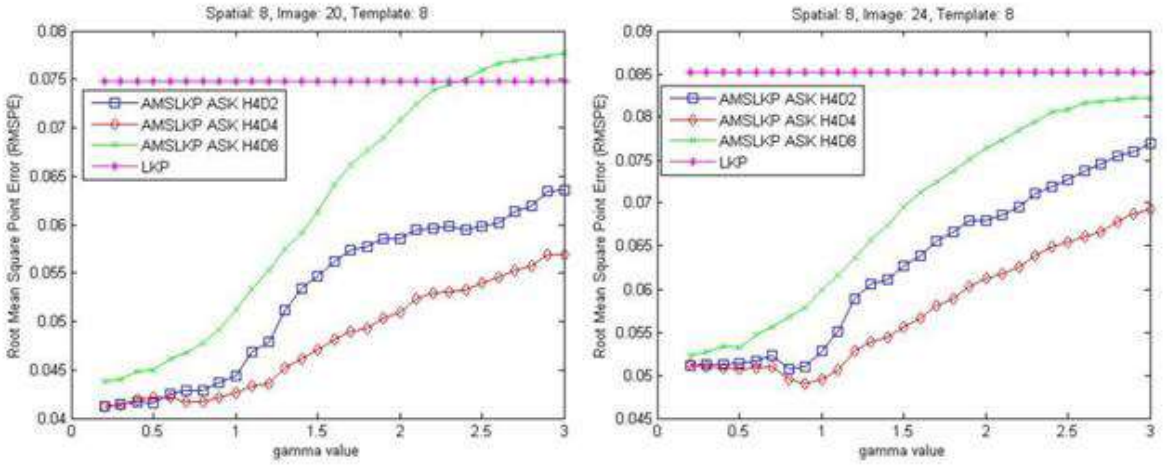
4.6 Convergence Results

Applying all above settings, Figure 11 gives a more intuitive description of how the AMSLKP algorithm converges both between and within levels in the pyramid. Recall the parameter set used for AMSLKP is: $h_j \in \{1, 2, 3, 4\}$, $\theta_i \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, $\Gamma = 0.6$ and kernel type fixed to ASK. We presents here the results for both scenarios when $\sigma_s = 6, \sigma_I = 32, \sigma_T = 0$ and $\sigma_s = 8, \sigma_I = 24, \sigma_T = 8$.

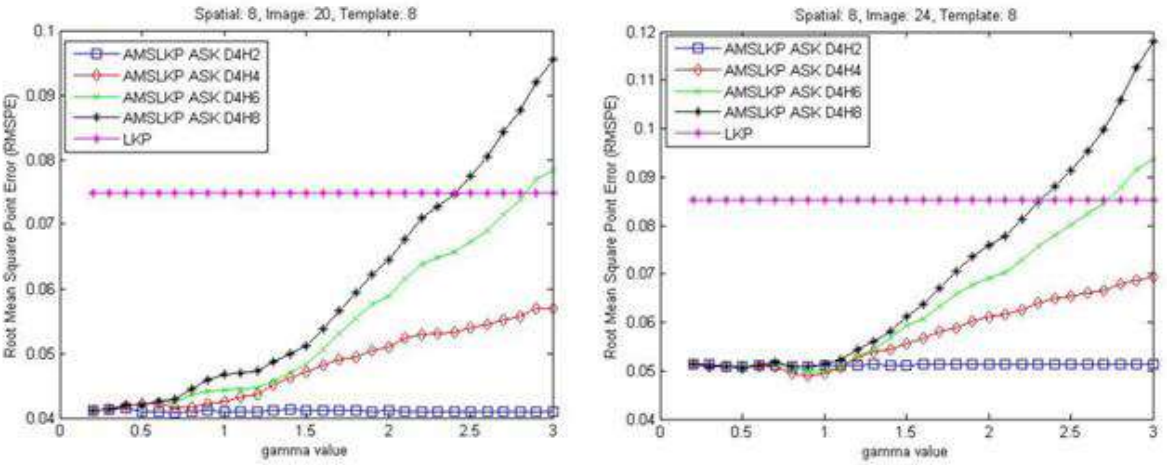
Totally, for all above noise conditions evaluated over the randomly chosen 10 FERET faces, our proposed AMSLKP method gains more than 40% accuracy over LKP method. Figure 11.a gives the all-level convergence results while Figure 11.b gives the corresponding 0^{th} level convergence result. In Figure 11.a, to visually illustrate the gradual inter-level converging process, we put all three level results together. Specifically, iteration 1-5 stands for 2^{nd} level, iteration 6-10 stands for 1^{st} level, while the last 5 iterations stand for 0^{th} level. Clearly, considering the parameter estimation accuracy, our gain over LKP method is obvious.



a. Results evaluated using different kernels



b. Results evaluated using different sector divisions



c. Results evaluated using different scales

Figure 10 Scenario 2: RMSPE- Γ performance comparison between various kernel types, various kernel directions and different kernel scale sets

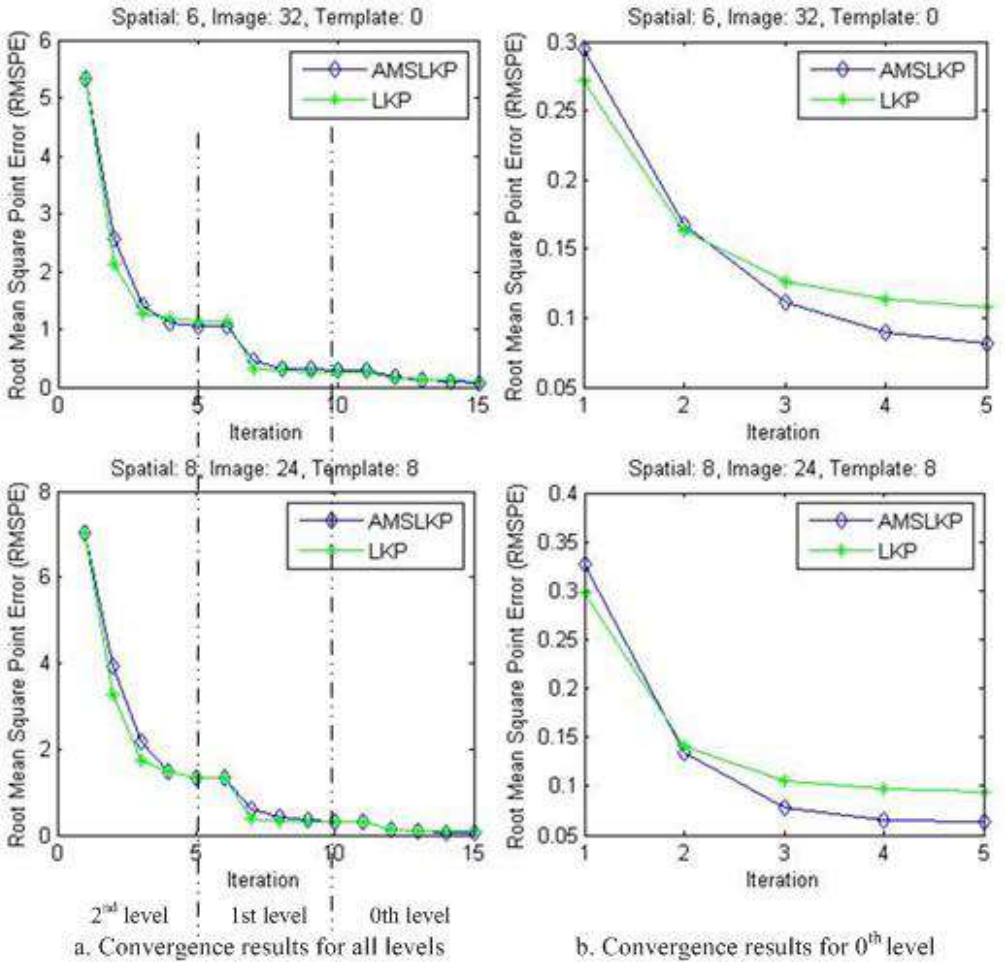


Figure 11 Convergence Results

5. CONCLUSION

This paper proposes a novel Anisotropic Multi-Scale Lucas-Kanade Pyramid (AMSLKP) method which embeds the Anisotropic LPA-ICI Filter in the pyramid LK structure. Instead of calculating gradients in only one direction with fixed scale, we utilize multiple directions with multiple scale sizes and select the optimum scale adaptively for each point in the image. Besides, the estimate of the noisy image is also calculated and then utilized in the pyramid LK process. Under all kinds of noise conditions, the experiment result shows that the proposed AMSLKP method improves the parameter accuracy by more than 40% compared to Lucas Kanade pyramid.

REFERENCES

[1] Lucas, B., Kanade, T., "An iterative image registration technique with an application to stereo vision," In Proceedings of the International Joint Conference on Artificial Intelligence, 674-679(1981).

[2] Baker, S., Gross, R., Matthews, I., and Ishikawa, T., "Lucas-Kanade 20 years on: A unifying framework: Part 1," Technical Report CMU-RI-TR, 03(01),(2003).

[3] Cootes, T., Edwards, G., and Taylor, C., "Active Appearance Models," IEEE Trans. Pattern Analysis and Machine Intelligence, 23(6), 681-685 (2001).

[4] Nicholas, D., Richard, B., "Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation," IEEE Transaction on Pattern Analysis and Machine Intelligence, 3(1), (2008)

[5] Katkovnik, V., Egiazarian, K., Astola, J., [Local Approximation Techniques in Signal and Image Processing], SPIE Proc., Monograph Vol. PM157, 173-261 (2006).
[6] Bouguet, JY., "Pyramid Implementation of Lucas Kanade Feature Tracker: Description of the algorithm," http://robots.stanford.edu/cs223b04/algo_tracking.pdf, (2009)
[7] Bouguet JY, "Pyramid Implementation of the Affine Lucas Kanade Feature Tracker: Description of the algorithm", http://robots.stanford.edu/cs223b04/algo_affine_tracking.pdf, (2009).
[8] Katkovnik, V., Egiazarian, K., Astola, J., "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule," J. of Math. Imaging and Vision. 16, 223-235 (2002).
[9] Katkovnik, V., Egiazarian K., Astola J., "A Spatially Adaptive Nonparametric Regression Image Deblurring", IEEE Trans. Image Process. 14(10), 1469-1478 (2005).
[10] Katkovnik, V., Foi, A., Egiazarian, K., Astola, J., "Directional Varying Scale Approximation for Anisotropic Signal Processing," EUSIPCO Proc. of XII European Signal Process. Conf., 101-104 (2004).
[11] Katkovnik, V., Astola,J., Egiazarian, K., "Phase local approximation (PhaseLa) technique for phase unwrap from noisy data," IEEE Trans. Image Process., 17, 833-846(2008).
[12] Katkovnik, V., Paliy, D., Egiazarian, K., Astola, J., "Frequency domain blind deconvolution in multiframe imaging using anisotropic spatially-adaptive denoising," Proc. 14th European Signal Process. Conf., (2006).
[13] Paliy, D., Katkovnik, V., Bilcu, R., Alenius, S., Egiazarian, K., "Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data," International Journal of Imaging Systems and Technology (IJISP), 17, 105-122(2007).
[14] Philips P., Moon H., Pauss P., and Rivzvi S., "The FERET Evaluation Methodology for Face-Recognition Algorithms", In Proceedings of CVPR,137-143 (1997).
[15] Foi, A., "Anisotropic nonparametric image processing: theory, algorithms and applications," Ph.D. Thesis. Dip. di Matematica Politecnico di Milano, (2005).

Publication II: Face recognition using scale-adaptive directional and textural features. Mehta R, Yuan J, Egiazarian K. Pattern Recognition, 2014, 47(5): 1846-1858.

Pattern Recognition 47 (2014) 1846–1858

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Face recognition using scale-adaptive directional and textural features

Rakesh Mehta*, Jirui Yuan, Karen Egiazarian

Tampere University of Technology, P.O. Box 527, FI-33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 4 October 2012
Received in revised form
9 October 2013
Accepted 13 November 2013
Available online 22 November 2013

Keywords:
Face classification
Face representation
Local Polynomial Approximation (LPA)
Local Binary Patterns (LBP)

ABSTRACT

A novel approach to face recognition problem using directional and texture information from face images, is proposed in this paper. In order to capture the directionality, specially designed using local polynomial approximation technique, scale adaptive digital filters are used. For texture features extraction, a low dimensional and computationally effective local descriptor is utilized. Textural and directional features are captured at the holistic and part based levels resulting in a robust face descriptor. The proposed method is tested on a number of standard test face datasets (ORL, XM2VTS, Extended Yale, CMU-PIE, AR, and FERET) for different scenarios and its performance is compared with several state-of-the-art techniques.

1. Introduction

A main goal of any face recognition system is to automatically identify a person from a given image or a sequence of images or video. There are many such systems used in various commercial, security and surveillance applications. Although the task of face recognition and identification is very simple and natural for human beings, for an automated system it is quite a challenging task. The challenges an automatic face recognition system may encounter are: the variations in face images due to different illumination conditions, pose changes, occlusions, aging aspects, poor image quality due to the usage of low cost camera, and so on. These problems make face recognition an active field of research both in computer vision and pattern recognition.

The automatic face recognition task can be broadly divided into three main parts: detection, representation and classification. Face detection involves localizing the face in an image. Representation deals with the extraction of specific features from the face images that accentuate certain aspects. Classification refers to the decision making process where an unlabeled face image is classified into a specific class. Face representation plays a critical role in the overall performance of a face recognition system. A robust face representation should be discriminative for different individuals and invariant to external changes such as illumination and other variations mentioned above. This paper addresses the problem of representation where we assume that a face has been localized using a face detection algorithm and alignment has been done by rotating and scaling the face image. Hence we consider that normalized and cropped frontal face images have been acquired.

In this work we use the directionality and the texture of face images at the global and local levels for face representation. Texture is a discriminative cue and has been extensively utilized for face recognition using filter banks and local descriptors [1–6]. Directionality is a low level shape feature that enchances edges and boundaries and provides information that is complimentary to texture. We consider a combination of these features to extract discriminative information from the face image. The directionality of the facial features is captured using the scale based directional derivatives and the textural features are extracted by applying a low dimensional and easy to compute local descriptor. These features are captured at both, local and global levels, to generate a robust face descriptor. The robustness of the proposed method is evaluated by performing the tests in different scenarios such as: (1) illumination variation, (2) expression variation, (3) occlusions (sunglasses and scarf), (4) large number of subjects, (5) Gaussian noise and (6) missing pixels from image (negative impulse noise). The results are compared with the existing state-of-the-art methods.

The remaining part of the paper is organized as follows. Section 2 briefly presents a background of the related work. In Section 3, the details of the proposed face representation approach based on directional and textural features are described. The performance and computational time of the proposed algorithm critically depend on several parameters that are discussed in Section 4. Experimental results and analysis is exhibited in Section 5 and Section 6 concludes the paper.

2. Background

Over the past few decades many different approaches have been developed for face representation. A class of methods, called

* Correspondence to: TC 403, Department of Signal Processing, Tampere University of Technology, FI-33101 Tampere, Finland.
Tel.: +358 4 0366 4084; fax: +358 3 364 1352.
E-mail addresses: rakesh.mehta@tut.fi (R. Mehta), jirui.yuan@tut.fi (J. Yuan), karen.egiazarian@tut.fi (K. Egiazarian).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.11.013

holistic, considers a face image as a whole and extracts the global features from it. The most popular of the holistic techniques are Eigenface [7] and Fisherface [8] which are based on Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), respectively. The first method projects the image into a subspace where the individual components are ranked according to their variances. The face image is reconstructed by a linear combination of these components that satisfy the least mean square criterion. One of the significant drawbacks of Eigenface based representation is that it is highly sensitive to illumination and pose changes. To address this problem, Belhumeur et al. proposed Fisherface method [8] which maximizes the ratio of between-class-variance and within-class-variance. Although the Fisherface based recognition outperforms Eigenface with sufficient training samples, the efficiency of first method reduces significantly as the number of training samples decrease [9]. The problem of low training samples is addressed in [10,11]; however these algorithms discard the discriminative information in null or principle space. To fill this gap, Yang et al. proposed Complete LDA (CLDA) [12] utilizing the complete discriminative information in the null and the principle space. Recently, Lu et al. developed a new algorithm for CLDA with an incremental learning [13].

Since a face image is spatially varying, significant information is not used by the holistic face representation. To capture the spatial structure, part based face representation methods have been proposed where different regions of face images are encoded separately. Samaria et al. [14] applied Hidden Markov Model (HMM) for face recognition by dividing the face image into overlapping horizontal segments. In their method, a model is created by segmenting the face image from the top to the bottom into separate sections corresponding to forehead, eyes, nose, mouth and chin. Wiskott et al. [15] proposed a part based representation based on the Dynamic Link Archetecture (DLA) and Gabor jets. This algorithm represents a face as labeled graphs whose nodes correspond to key points of the face such as pupils, tip of nose, corner of mouth, etc. It achieved high classification accuracy at the expense of computational complexity. Among the recent part based face recognition schemes Local Binary Patterns (LBP) [16] has gained popularity because of its computational simplicity. LBP operator generates a series of binary codes based on the signs of pixel derivatives with respect to its neighbors. Ahonen et al. applied LBP to face recognition [17] by partitioning the face image into non-overlapping rectangular blocks. Xiaoyang et al. [18], extended the idea of LBP to Local Ternary Pattern (LTP). LTP considers the magnitude of pixel derivatives along with its sign to generate ternary code. Zhang et al. proposed Local Derivation Pattern (LDP) [19], which computes the higher order derivatives of a pixel with respect to its neighbors. The local descriptors based methods (LBP, LDP, and LTP) achieve invariance to rotation and monotone transformation to certain extent and have shown quite promising results.

The hybrid approaches use both local and global features of the face image. Pentland et al. [20] extended the idea of Eigenfaces to specific Eigenfeatures called Eigenmouth, Eigeneyes, etc. The combined representation of Eigenface and Eigenfeatures achieved better performance than the Eigenface based approach. Blanz et al. [21] proposed a hybrid approach in which firstly the face image is decomposed into the different sections corresponding to facial features such as mouth and eyes. Then, a 3D model of face is used to generate face images under different illumination conditions and poses to train the classifier. In a number of approaches [1–6] part based face recognition techniques is applied after processing the face image in a holistic manner. For example, Zhang et al. [1] applied a set of Gabor filters on face images followed by block processing using LBP. However, the inter-scale dependences between various Gabor filtered images is not considered by this approach. Lei et al. [2] proposed Gabor Volume based LBP (GV-LBP) to address this problem. GV-LBP captures inter-scale dependences between adjacent scales and orientations of Gabor face images. These approaches [1–6] generally apply a set of filters leading to 40 Gaborfaces. Block processing of these Gaborfaces results in a very high dimensional face descriptor. Thus, Gabor filters based approaches are computationally expensive even when the image size is relatively small. Li et al. [22], proposed the use of heat kernels in order to extract the structural information from the face image followed by a texture extraction using LBP. This method achieved better results than the ones using the combination of Gabor and LBP, at the expense of complex mathematical operations on large matrices.

The face representation method proposed in this paper extracts directional and textural features. Directional features are extracted at the multiple scales by convolving the face image with a set of Local Polynomial Approximation (LPA) filters [23]. Then the scales are adaptively optimized by applying Intersection of Confidence Interval (ICI) rule to obtain Optimized Directional Faces (ODFs). To capture the textural features a modified LBP operator [24] is applied on the ODFs. Finally, LDA is used to reduce the dimensionality of the face descriptor and a Support Vector Machine (SVM) [25] classifier is trained in the descriptor space to perform the face classification.

3. Face representation based on directionality and texture

In this section, we describe the algorithm for facial representation based on directionality and texture. First, we introduce Optimized Directional Faces (ODFs) that capture the directional features from the face image at multiple scales. Then, we discuss the technique to incorporate both, global and local textural features in the face descriptor. Since a large number of local features results in high dimensionality, a LDA based dimensionality reduction technique is examined to obtain a low dimensional face descriptor. Finally, we present the summary of the face representation algorithm.

3.1. Optimized directional faces (ODFs)

This section describes the extraction of the directional information from the face image using the image derivative. Directional features are considered because a substantial amount of information in face image is aligned in specific directions. Image derivatives are used because they can enhance the facial features that carry very characteristic information about a person. In our approach we use LPA to compute the directional derivatives, as these are specifically designed for directional or anisotrophic image analysis. Furthermore, when combined with ICI it provides image derivatives that are based on optimum scale, and therefore highlights the details in fine manner. The LPA [23,26–28,42] is a general tool for linear filter design, in particular, for a design of the directional filters at multiple scales. It is based on the idea that a function is well approximated by a set of linearly independent polynomial in the neighborhood of point of interest x. The first order LPA derivative filter ($g^1_{s,\theta}$) for direction $\theta \in \Theta$ and scale $s \in S$ is defined as follows:

$$g^1_{s,\theta}(x-v) = -w_{s,\theta}(x-v)\phi^{\mathrm{T}}(x-v)\Phi_s^{-1}\phi^1(0), \tag{1}$$

$$\Phi_s = \sum_v w_{s,\theta}(x-v)\phi(x-v)\phi^{\mathrm{T}}(x-v). \tag{2}$$

where $v \in R^2$ is a domain of coordinates, $x \in R^2$ is the center of the LPA, $w_{s,\theta}$ is a window function used for a fitting in the neighborhood of the center x, the scale s determines the size of the neighborhood, ϕ is a vector indicating the set of 2D linearly

independent polynomials. LPA is based on the idea that a function can be expressed as a sum of linearly independent polynomials ϕ in a sliding window $w_{s,\theta}$. Design of the window $w_{s,\theta}$ is explained in Appendix I and the parameter setting are given in Section 4.1.

The thorough analysis of technical and theoretical details concerning the LPA filters can be found in [42]. Here we discuss the scale s and the direction θ which are two important parameters in the LPA filters. The direction specifies the spatial orientation of the filter. We consider four directions $\theta \in \Theta = \{t\pi/4, t = 0, 1, \ldots, 3\}$ (horizontal, vertical and two diagonal) because these are aligned with the majority of the prominent facial features, the choice is also justified empirically in Section 4. The scale parameter, in context of LPA filters, refers to the support size of the filters. It determines the number of neighboring pixels considered to estimate the directional derivative at a point. It is an important parameter in case of noisy images, because large scale estimates are more robust to noise. The detail of selecting the scale is presented in Section 4, here we consider a general case.

For the face image $I(x)$, the direction $\theta \in \Theta = \{t\pi/4, t = 0, 1, 2, 3\}$ and the scale $s \in \{s_1, s_2, \ldots, s_J\}$ the directional face $(D_{s,\theta}(x))$ is defined as

$$D_{s,\theta}(x) = (I * g^1_{s,\theta})(x) = \sum_v I(v) g^1_{s,\theta}(x - v), \tag{3}$$

where $*$ indicates the 2D convolution operator. The number of resulting directional faces depends on the number of scales and directions used in LPA filters, as such:

$$\text{No. of directional faces} = \text{No. of directions} \times \text{No. of scales}. \tag{4}$$

Fig. 1 shows the 16 directional faces obtained using the four directions and four scales. It can be observed that the directional faces obtained using horizontal filters highlight horizontal facial features (like eyes, eyebrows and lips), while the directional faces obtained from vertical filters highlight vertical features (like nose and face contour), and the angular filters highlight the boundaries of these facial features.

In our previous work [30], we used directional faces directly to extract the face descriptor. Since four scales and four directions result in 16 directional faces, feature extraction from these directional faces leads to a high dimensionality and computational complexity. In this paper, the scale for each direction is optimized using ICI rule [23]. The ICI rule is an algorithm for adaptive selection of scale for each pixel. As we have the LPA estimates for a set of increasing scale values $s_1 < s_2 \cdots < s_J$, we use ICI to select the estimate which minimizes the MSE with respect to variation of scale. It determines a sequence of confidence intervals Q_i for each scale s_i [23,29]

$$Q_i = [D_{si,\theta}(x) + \Gamma\sigma_{D_{si,\theta}(x)}, D_{si,\theta}(x) - \Gamma\sigma_{D_{si,\theta}(x)}], \tag{5}$$

where Γ is the user specified threshold Gamma parameter, $\sigma^2_{D_{si,\theta}(x)} = \sigma^2 \sum_x g^2_{\theta,s_i}(x)$ and σ^2 is the variance of the noise present in the face image. As the scales increase, the standard-deviations decrease so does the confidence intervals. The ICI rule selects the estimate corresponding to the largest scale whose confidence interval intersects with the confidence interval of the smallest scale. Formally ICI rule states: *Consider the intersection of interval* $\mathcal{I}_j = \cap_{i=1}^{j} Q_i$, *where* Q_i *is defined in* (4) *and let* j^+ *be the largest of the indexes* j *for which* $\mathcal{I}_j$ *is non-empty,* $\mathcal{I}_{j^+} \neq \emptyset$ *and* $\mathcal{I}_{j^+ + 1} = \emptyset$. *Then adaptive scale* s^+ *is defined as* $s^+ = s_{j^+}$. Roughly speaking the rule defines the adaptive scale s^+ as largest one in S, the derivative estimate $(D_{s^+,\theta}(x))$ of which does not differ significantly from the derivatives corresponding to the smaller scales. This optimization of s for each of the directional estimates yields the adaptive scales s^+ for every pixel in direction θ. The directional derivatives obtained after optimization of scale are called Optimized Directional Faces (ODFs). Since four directions are used, $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$, we obtain four corresponding ODFs, $\{O_{\theta_j}\}_{j=1}^4$. ODFs for a face image are shown in Fig. 1.

3.2. Global and local textural features

To capture the textural features a LBP based local texture descriptor is applied on ODFs. LBP is frequently used in face recognition applications, due to its computational simplicity and effectiveness. When applied on local image regions it results in high dimensionality. In order to keep the dimensionality and complexity low, we apply a modified LBP (mLBP) [24] operator in a 3×3 neighborhood. Fig. 2 shows the notation used for 3×3 neighborhood pixel values and an illustrative example of mLBP operator. The operator is defined as

$$mLBP(t_c) = \sum_{i=0}^{3} f(t_i - t_{i+4}) 2^i + f(t_c - \bar{t}) 2^4 \tag{6}$$

where $f(t)$ denotes a step function, i.e. $f(t) = 1$ if $t \geq 0$, else $f(t) = 0$ t_c is intensity value of the central pixel, t_i are the neighbors around

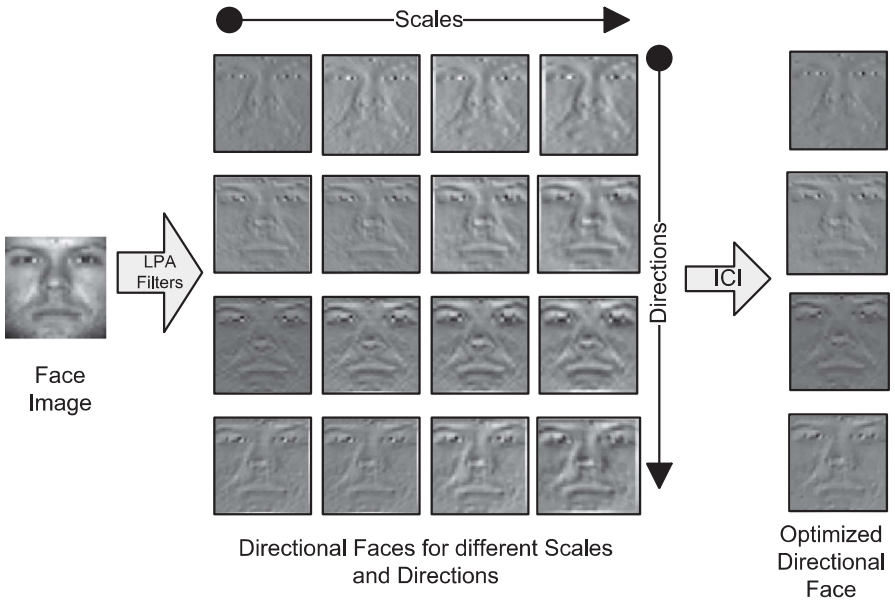


Fig. 1. The face image is convolved with the LPA filters to obtain Directional Faces at different scales and directions. The scale is optimized for each direction resulting in an Optimized Directional Face for each direction.

the central pixel and $\bar{t}=(t_0+t_1+\cdots+t_7)/8$ is the mean of the neighboring pixels. The mLBP operator utilizes 5 binary bits to encode any pixel t_c, which outputs a number between 0 and 2^5-1(0–31). Computing a histogram over these values results in a descriptor whose dimensionality is equal to 32. This dimensionality is eight times less than the one obtained by the original LBP.

The textural features are extracted from each ODF separately and concatenated with each other. The process of textural feature extraction from an ODF consists of three steps: (i) applying mLBP operator, (ii) partitioning of mLBP feature image and (iii) histograms computation and concatenation. As shown in Fig. 3, first, mLBP operator is applied on an ODF (O_{θ_j}) to extract low dimensional textural features. Then the mLBP feature image is partitioned at various levels to extract the spatial information. Levels of partitioning ($l=0, 1,\ldots, L$) are defined from the lowest level ($l=0$) that considers the whole image, to the higher levels ($l>0$) where image is partitioned into 4^l non-overlapping blocks. Histograms obtained from blocks of level l of O_{θ_j} are concatenated together and represented as $h_{j,l}$. While combining the features from different levels a unique weight v_l is associated with each level l. For each ODF, $\{O_{\theta_j}\}_{j=1}^{4}$, the features of different levels $h_{j,l}$ are concatenated using the unique weight v_l as

$$H_{\theta_j}=(v_0h_{j,0}, v_1h_{j,1}, \ldots, v_Lh_{j,L}), \quad v_l=\frac{1}{2^{L-l+1}}. \tag{7}$$

Similar procedure is done for all the four ODFs and the feature vectors are concatenated with each other $F=\{H_{\theta_1}, H_{\theta_2}, H_{\theta_3}, H_{\theta_4}\}$.

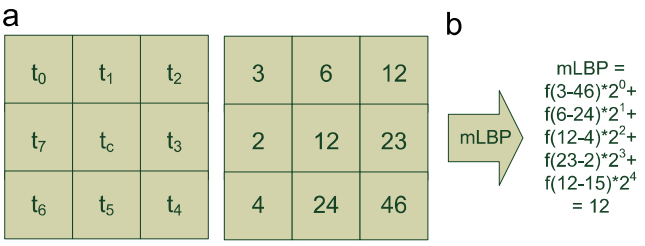


Fig. 2. (a)The notation for 3×3 neighborhood and (b) illustration for mLBP operator.

If we consider partitioning of the ODF at 4 different levels $\{l=0,1,2,3\}$, then the dimensionality of the descriptor is given by (no. of directions × dimensionality of mLBP × no. of image partitions) that is $4\times32\times\sum_{l=0}^{3}4^l=10{,}880$. This high dimensionality of the face descriptor is due to the image partitioning at high levels, such as $l=2,3$, where image is divided into small size segments. The proposed multilevel partitioning of the ODFs not only extracts the local features at different levels for $l>0$ but also preserves the holistic features of face image at level $l=0$. It also results in high dimensionality which is reduced by applying LDA technique as discussed next.

3.3. Dimensionality reduction and classification

Multilevel directional and textural feature extraction leads to high dimensionality. In order to reduce the dimensionality, the LDA based technique [8] is applied. The idea behind LDA is to project the feature vector on a subspace such that the vectors of the same class are grouped together, while the vectors of different classes are dispersed as much as possible.

Given a set of training feature vectors, $\{F_1,F_2,\ldots,F_m\}$ each belonging to one of the classes $\{c_1, c_2, \ldots, c_k\}$, the within class scatter matrix (S_W) is defined as

$$S_W=\sum_{i=1}^{k}\sum_{j\in c_i}(F_j-\bar{F}_i)(F_j-\bar{F}_i)^T, \tag{8}$$

where k is the total number of classes, $\bar{F}_i=(1/N_i)\sum_{j\in c_i}F_j$ is the mean of feature vectors of class c_i and N_i is the number of samples in class c_i. The measure of the spread of vectors among different classes is given by between class scatter matrix, which is defined as

$$S_B=\sum_{i=1}^{k}N_i(\bar{F}_i-\bar{\bar{F}}_i)(\bar{F}_i-\bar{\bar{F}})^T, \tag{9}$$

where $\bar{\bar{F}}$ is the mean of all the feature vectors. The optimum projection matrix $W_{opt}=[w_1, w_2, \ldots, w_{k-1}]$ which satisfies Fisher's criterion is defined as

$$W_{opt}=\arg\max_{W}\frac{|W^TS_BW|}{|W^TS_WW|}=[w_1, w_2, \ldots, w_{k-1}], \tag{10}$$

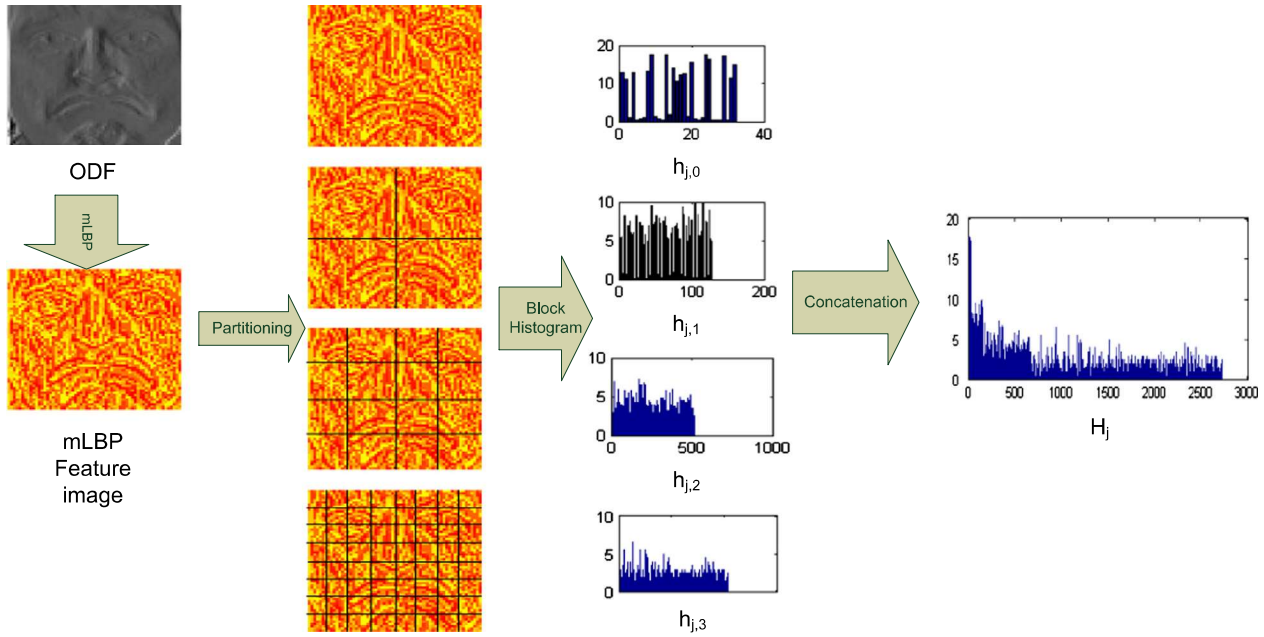


Fig. 3. The mLBP operator is applied on the Optimized Directional Face to obtain mLBP feature image. The mLBP feature image is partitioned at different levels ($l=0,1,2,3$) and at each level it is partitioned into $4l$ blocks. The histograms of the blocks at a level l are concatenated and represented as $h_{j,l}$. These histograms are further weighted for each level and concatenated.

where w_i for $(i=1,2,..,k-1)$ are eigenvectors of $S_W^{-1}S_B$ corresponding to $k-1$ non-zero eigenvalues. The upper bound on the generalized non-zero eigenvectors is $k-1$, hence we obtain the projection matrix as $W_{opt} \in R^{(k-1)\times D}$. The features vectors $F_i \in R^{D\times 1}$ are multiplied with W_{opt} that changes the dimensionality of the vectors from D to $k-1$, which is much less than D. This step of the algorithm also increases the classification accuracy because the class specific information is used while projecting the feature vectors into the new feature space.

Finally, after capturing the information at multiple levels and reducing the dimensionality, the face image is represented in a low dimensional space. Face classification is performed using an SVM classifier. The SVM is computationally faster than nearest neighbor classifier. A multiclass SVM classifier with linear kernels is trained by using the libSVM tool [25]. Linear kernels use the features directly; thus the improvement in the classification results can be attributed to the used features.

3.4. Face representation algorithm

The proposed face representation algorithm is summarized in Fig. 4.

4. Implementation and parameters setting

In this section we present the implementation details of LPA filters and clarify the choice of critical parameters used in the algorithm. The parameters discussed are (1) number of scales and directions for directional filters, (2) Gamma parameter (Γ) in ICI rule and (3) number of partitioning levels. The effect of these parameters is studied empirically by performing experiments on face images from the standard dataset (Extended YaleB).

4.1. LPA implementation

LPA assumes that a function can be approximated by a set of polynomials ϕ as mentioned in Section 3. The set ϕ is defined by the polynomials entries $x_1^{k_1}x_2^{k_2}/k_1!k_2!$. The number of polynomial used for approximation are restricted by the order of the LPA, m, such that $0 \le k_1+k_2 \le m$, $k_1, k_2 \ge 0$. Linearly independent polynomials are obtained for different values of exponent sum (k_1+k_2). For examples, if $k_1+k_2=0$ we get the $\phi_0=1$ and for $k_1+k_2=1$ the polynomials are $\phi_1=x_1$ and $\phi_2=x_2$. In our experiments the order of the LPA is set to $m=1$, as higher order results in noiser estimates, thus the set of linearly independent polynomial becomes $\phi=[1,x_1,x_2]^T$.

4.2. Number of scales and directions

The scale in the context of directional filters, represents the size of the window function and, therefore, determines the size of filters. If it is too large, it prescribes equal weights to all residuals and results in oversmoothing. If the scale is of the minimal size, then the estimate is the same as the point of observation. To find a sufficient number of scales, these are varied in our experiment from 2 to 5. As more scales are used, more directional information from the face images is extracted. If the number of scales is too large, then higher scales do not yield a robust estimate and the accuracy decreases. The results for a classification for various scales are shown in Fig. 5. The average accuracies for the scales were empirically found to be (78.91, 81.25, 83.16 and 82.50). It can be observed that the accuracy increases till a particular point after which it starts decreasing.

To generate the directional kernel for a particular direction θ_i, we need to rotate the window w_s by angle an θ_i in order to direct it to the desired direction. Derivatives in different directions reflect diverse directional information of a face image. Directions are

Input: Face image *I(x)* *Output*: Face descriptor

1) Apply a set of filters $g^1_{s,\theta}$ on the input face image *I(x)* to generate directional faces $(D_{s,\theta}(x))$ corresponding to four scales and four directions.
2) Optimize the scale in each direction using ICI rule to obtain four ODFs, $\{O_{\theta_j}\}_{j=1}^4$.
3) For each ODF, $\{O_{\theta_j}\}_{j=1}^4$, repeat steps a-c
   a. Apply mLBP operator on O_{θ_j}.
   b. For partitioning level l = 0 to L do following
      i. Partition mLBP feature image into 4^l rectangular blocks.
      ii. Compute the histogram for each block.
      iii. Concatenate the histograms of all blocks at current level to form vector $h_{j,l}$.
   c. Combine the histogram of different levels to obtain H_{θ_j} as mentioned in (6).
4) Concatenate the features corresponding to four ODFs as $F=\{H_{\theta_1}, H_{\theta_2}, H_{\theta_3}, H_{\theta_4}\}$.
5) Project the feature vector using the optimum projection matrix W_{opt} to obtain the face descriptor.

Fig. 4. The algorithm for the proposed method.

chosen by uniformly sampling the interval $[0, \pi]$ and by increasing the number of samples. Thus the directions in our experiment are defined as $\theta_i = \{(i\pi/N),\ i = 0, 1, \ldots, N-1\}$, where N is the total number of directions considered.Table 1 shows the change in the accuracy with the increase in the number of directions. It can be observed that the accuracy increases initially after which starts decreasing. The best accuracy is obtained for four directions, thus the directions correspond to $\{0, \pi/4, \pi/2, 3\pi/4\}$.

4.3. Gamma parameter for ICI

Gamma is an important parameter in ICI rule. It determines the threshold of the confidence interval. The width of the confidence interval, for a scale s and direction θ, is given by $2\,\Gamma\sigma_{D_{s,\theta}}$, which depends only on Γ and a standard deviation of an estimate of the image. The optimum value of gamma for a set of images depends on the amount of noise present in these images. If the noise is not present then the smaller value of gamma is to be used, but if the noise exists in face images then higher values of gamma are required. Here we study the effect of noise on the "optimum" value of gamma. The value of gamma is varied from 0 to 4 at stepsize of 0.2 and three different noisy scenarios are considered with additive Gaussian noise of zero mean and $\sigma = 4$, 8 and 12. When no noise is added to the face images, i.e. $\sigma = 0$, the optimum value of gamma is found to be 0.2. Small value of gamma implies that the confidence interval between consecutive scales is small and thus the estimate computed by one of the smallest scale is selected. When noise is added with $\sigma = 4$, it can be seen that high accuracy is achieved at larger values of gamma. For $\sigma = 8$ and 12 it can also be observed that the accuracy increases as the gamma increase and then it becomes almost constant. The classification accuracy for $\sigma = 4$, 8 and 12 with respect to gamma is shown in Fig. 6. Thus, it can be seen that when noise is present the optimum value of gamma lies between 2 and 2.5.

4.4. Partitioning levels

The level of partitioning is an important factor which maintains a balance between the dimensionality and the extraction of part based features. With higher partitioning level, there is an increase in the number of blocks resulting in higher dimensionality. Computational complexity also increases with the partitioning level. Table 2 shows the classification performance as the levels increase with respect to the time. The time is shown for the feature extraction and classification of 2224 test face images of YaleB dataset which are trained using 5 training samples for each individual. The experiment is performed on a Windows 7 machine with 4 GB RAM and Intel core2duo processor. The lower levels encode the holistic information of the face while the higher level encodes the part based information. Fig. 7 shows the histogram obtained for the first three levels of partitioning. It can be observed that the zeroth level is similar even for different class of samples. It contains the information of the general structure of the faces and, thus, does not vary much across different classes.

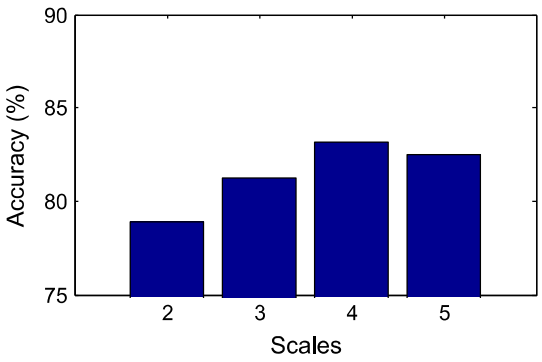


Fig. 5. The change in the classification accuracy with respect to a number of scales used.

Table 1
Classification accuracy with increasing number of directions.

Directions	1	2	3	4	5	6	7	8
Accuracy(%)	87.66	89.33	91.10	93.61	92.60	92.70	92.54	91.68

5. Experiments

To evaluate the performance of the proposed algorithm experimental tests are conducted on a number of publicly available face datasets. In this section we first describe these test datasets and the experimental conditions. It follows a detailed comparison of the proposed algorithm with the state-of-the-art methods and a subjective performance evaluation.

The datasets used in our experiments are ORL [43], XM2VTS [31], YaleB [32], extended YaleB [33], CMU-PIE [34], AR [35] and FERET [36,37].

ORL dataset consists of images taken from 40 different individuals with 10 images of each person. The images were taken at different times, varying the lighting, facial expressions (open/closed

Table 2
Computational complexity with increase in partitioning levels.

Partitioning levels (l)	0	1	2	3
Time (s)	16.63	28.60	61.07	174.59
Accuracy (%)	3.77	34.48	77.65	95.14

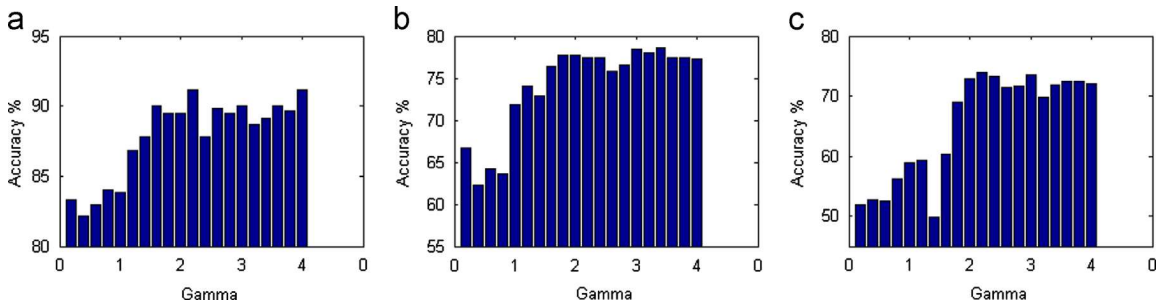


Fig. 6. Variation of classification accuracy with respect to Gamma for different noise levels. Three different noise levels are used with: (a) $\sigma = 4$, (b) $\sigma = 8$, and (c) $\sigma = 12$.

eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with a tolerance for some side movement).

XM2VTS dataset contains 2360 face images taken from 295 subjects varying in poses, hairstyles, expressions and glasses. There are eight shots of each person, obtained from four sessions spread out in monthly intervals. Compared to the other datasets it has high number of subjects.

Extended Yale-B dataset consists of 16,128 images of 28 individuals and YaleB dataset consists of 5760 face images of ten individuals. The images are taken under 9 different pose and 64 different illumination conditions. In our experiment we use the images with frontal pose of YaleB and extended YaleB with all 64 different illumination settings. These two datasets capture large variation in illumination and are commonly used to study the effect of illumination on face recognition. The example images are shown in Fig. 8.

CMU PIE dataset consists of 41,368 images of 68 people. Each person is captured under 13 different poses, 43 illumination conditions, and 4 expressions. The dataset has extreme illumination variations and a large number of images for each subject.

AR dataset consists of more than 4000 images of 126 individuals of which 70 are males and 56 are females. In AR dataset the images were taken in two sessions separated by 2 weeks, considering expression (neutral, smile, anger and scream) and occlusion (sunglass and scarf) variations. In our experiments we take images of 45 men and 45 women. The example images from one of the sessions are shown in Fig. 9.

FERET dataset contains frontal face images divided into five categories: Fa, Fb, Fc, Dup1, and Dup2. Fb images were taken at the same day as Fa images and with the same camera and illumination condition. Fc images were taken at the same day as Fa pictures but with different cameras and illumination. Dup1 images were taken on different days than Fa but within a year. Dup2 images were taken at least one year later than Fa. In the FERET tests, 1196 Fa face images are gallery samples for which the labels are known. 1195 Fb, 194 Fc, 722 Dup1, and 234 Dup2 pictures are the probe set for which the labels are not known. The number of images in Fa set indicates the number of classes, so for each person there is just one training image.

Each of these datasets simulates a scenario which causes variation in the face image in real time face recognition application. Using these datasets we evaluate the performance of proposed method in different test scenarios such as, expression variation, illumination variation, occlusion, gaussian noise corrupted face images and face images with missing pixels. The experiments are divided into two main subsections: noise-free images and noisy images. In noise-free face images are used without adding any noise and dataset cover the tests scenarios as follows:

1 ORL dataset (expressions and illumination)
2 XM2VTS (large number of subjects)
3 Extended YaleB dataset (illumination)
4 CMU PIE (extreme illumination over large test set)
5 AR Dataset (occlusions and expressions)
6 FERET (large number of subjects)

For the noisy case, the images of the dataset are corrupted with two different types of noise:

1 Gaussian noise
2 Negative impulse noise

For all the tests the images are aligned using the eye position resized to 64×64 pixels unless specified otherwise. Throughout the paper we use classification accuracy as the evalutation term.

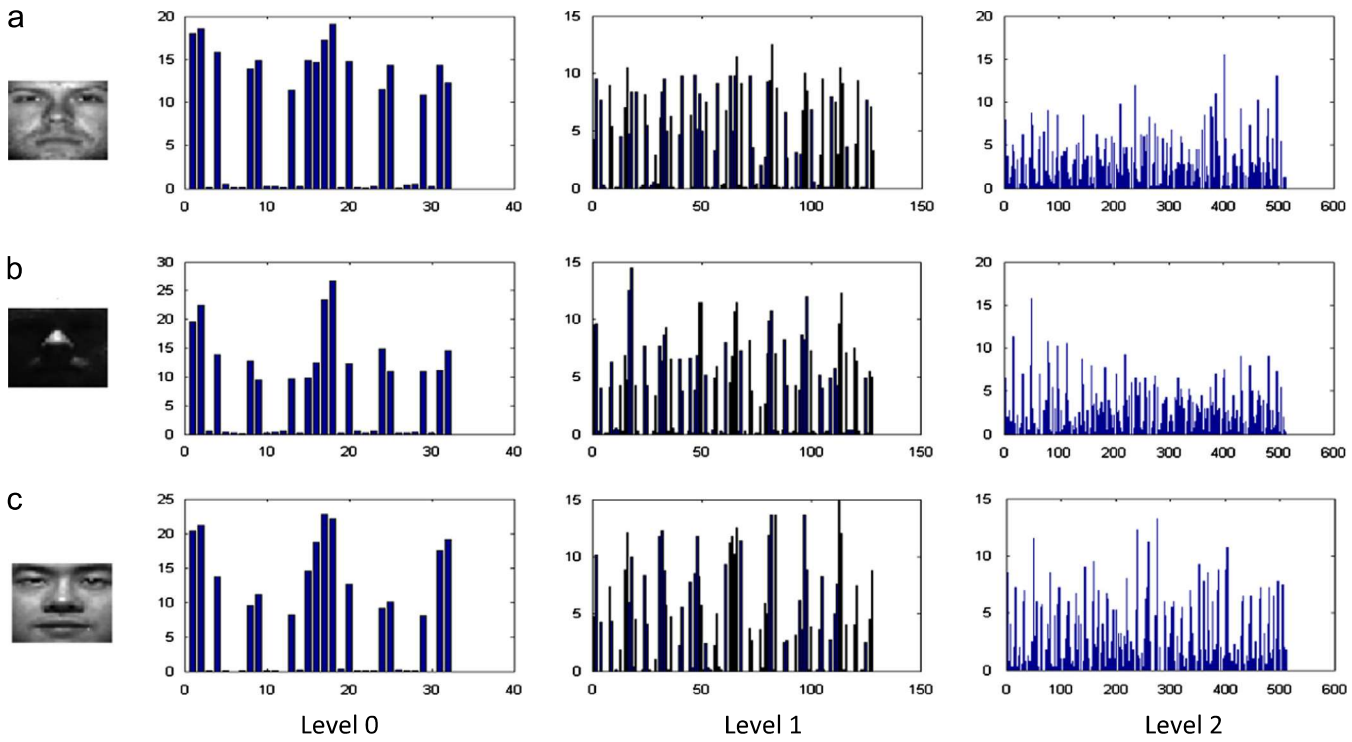


Fig. 7. Face images and the histograms obtained for different levels (l=0,1,2). Images (a) and (b) are of same subject but under different illumination condition while image (c) is of a different subject but with the same illumination condition as in (a). The *X*-axis of the histogram corresponds to the patterns obtained from mLBP and the *Y*-axis indicates the number of pixel. It can be observed that the pattern distribution of the histograms at level 0 is quite similar for all three images. This level encodes the general structure of the face while the higher levels 1 and 2 account for much finer details of the face image.

Fig. 8. Sample face images of a subject from Extended YaleB dataset.

Fig. 9. Sample face images from AR dataset. First row shows the expression variation, second show the scarf occlusion and third shows the sunglasses occlusion.

It is defined as the percentage of the test image correctly classified by the algorithm.

5.1. Noise free face images

In this section we present results of the experiments with noise free images. As it was discussed in Section 4, in this case the value of gamma for ICI is set to 0.2 and a linear SVM kernel is used for classification.

5.1.1. ORL dataset

For this dataset we follow the protocol similar to [12]. The number of training samples was set to 4 to evaluate the performance with less number of samples on a smaller dataset. Tests were conducted 50 times by randomly selecting the training samples from the dataset. The final accuracy is the average accuracy obtained for these 50 tests. The results are compared with Eigenface, Fisherface, LBP, LGBPHS [1] and GVLBP [2]. The accuracies of these methods and ours are 84.46, 91.16, 93.63, 94.13, 96.15 and 98.13, respectively. The comparison is shown in Fig. 10. For the small dataset like ORL it can be observed that, in general, a

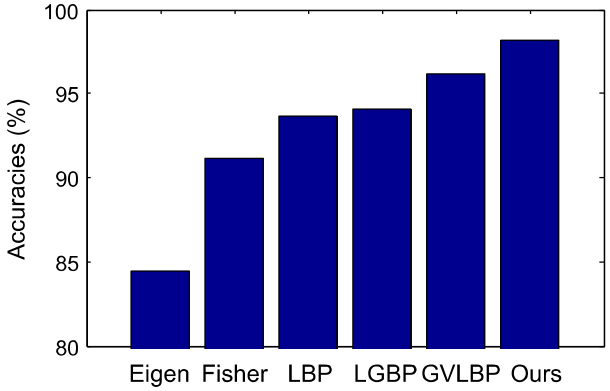


Fig. 10. The classification accuracies of different methods for ORL dataset.

Table 3
The dimensionality and computation time of descriptors.

Methods	Time (s)	Dimensionality
PCA	0.067	159
LDA	0.022	39
LBP	1.842	16,384
LGBPHS	89.29	655,360
GVLBP	95.53	655,360
Proposed	18.82	39

Table 4
XM2VTS dataset.

Methods	PCA	LDA	LBP	[38]	[39]	Ours
Accuracy (%)	88.81	79.66	96.61	95.00	**99.66**	**99.66**

high accuracy is achieved by all methods because the number of testing samples is not very high.

We also compare the computational complexity of the proposed method with the competitive methods. The time required to recognize a face image depends on two factors: computation time and dimensionality of the face descriptor. Table 3 shows the dimensionality of descriptors and the time required to classify the images from ORL dataset using various methods (PCA, LDA, LBP, LGBPHS, GVLBP). The images are partitioned into 8×8 blocks for LBP face descriptor and into 7×7 blocks for LGBPHS and GVLBP. For Gabor based methods we use filters at five different scales and eight orientations as mentioned in [1,2]. It can be seen that the subspace methods such as PCA and LDA have the advantage in terms of computation simplicity compared to other methods. However their performance is also much inferior to the proposed and other state-of-the-art methods. We also compare the performance of the proposed method with the LBP and the combination of Gabor and LBP (Gabor+LBP). It can be observed that both the computation time and the dimensionality of these face descriptors is very high not only compared to subspace methods but also compared to the proposed approach. The reason behind is that the Gabor based methods apply a filter bank of 40 filters which increases the computational time of feature extraction and also leads to high dimensional descriptors. For the proposed method it can be seen that the feature extraction time is higher than the subspace techniques but is much less than the Gabor+LBP based methods. However, the dimensionality of the proposed descriptor is comparable to the subspace methods which make the classification task fast. The experiment on computation time is performed on a Windows 7 machine with 4 GB RAM and Intel core2duo processor. (A demo code for proposed algorithm is available at http://www.cs.tut.fi/~mehta/demo_face.zip.)

5.1.2. XM2VTS dataset

For the XM2VTS database, we select four face images corresponding to four different sessions for all 295 subjects. Following the protocol of Tan and Yan [38] the images from the first three sessions are used as training data. The images from the fourth session for all the subjects are used as the testing data. With 295 subjects, this dataset tests the scenario for a relatively large number of test classes. The results are compared with PCA, LDA, LBP [38,39]. Table 4 shows the classification accuracies of these methods. Proposed algorithm gives a perfect result with nearly 100% accuracy of recognition.

5.1.3. Extended YaleB dataset

With the tests on a larger dataset the effectiveness of the proposed approach becomes even more evident in comparison with other approaches. In order to test the robustness to illumination variations YaleB and extended YaleB datasets are combined together. Four different tests are performed by changing the number of training samples. The number of training samples is set to 5, 10, 20 and 30. The results are compared with the Eigenface, Fisherface, LBP, CDA [40] and CAE [41]. Table 5 shows the classification accuracy for different number of training samples. It can be observed that the proposed algorithm outperforms the other approaches. Even with 5 training samples the proposed method results in a significantly high accuracy of 93.64% for largely varying illumination conditions, while the recent techniques achieve less than 80% of recognition accuracy. It can be observed that the performance of LBP degrades significantly with less number of training samples; it achieves an accuracy of 50% with 5 training samples. Thus, it is not able to capture the class specific features in illumination variation with less number of training samples. With 10 training samples the accuracy of the proposed method reaches almost 99%, which is significantly higher than the other methods.

Table 5
Classification accuracy for Extended Yale B Dataset.

	5 Trains	10 Trains	20 Trains	30 Trains
Eigenface	45.27	63.94	68.78	72.29
Fisherface	63.87	77.80	84.94	78.53
LBP	50.35	73.93	89.69	95.21
CDA	73.06	89.81	88.34	87.46
CEA	77.39	92.45	93.78	96.44
Ours	**93.64**	**98.84**	**99.54**	**99.64**

Table 6
Classification accuracy for CMU PIE dataset.

	5 Trains	10 Trains	20 Trains
Eigenface	39.19	66.18	73.24
Fisherface	56.25	70.01	78.09
LBP	55.87	77.32	90.17
CDA	59.03	78.73	89.60
CEA	64.71	81.13	90.55
Ours	**87.30**	**95.85**	**98.14**

The near invariance to illumination is due to the fact that the directional faces are the estimates of the directional derivatives; they are independent of the overall brightness of the face image. These estimates preserve the local edges and boundaries which are oriented in specific directions. These edges, boundaries and face structure provide the discriminative information which is used for face recognition.

5.1.4. CMU PIE dataset

The CMU PIE dataset consists of large number of images for each subject with extreme illumination conditions. For our experiments we have selected 11,556 frontal face images and performed the tests by varying the number of training images. Our test protocol is similar to Yang and Yang [12] in which all the 11,556 frontal face images from dataset are used, however we varied the number of training samples. The number of training samples is set to 5, 10 and 20 in subsequent experiments while the rest of the images are used as test data. Eigenface, Fisherface, LBP, CDA [40] and CAE [41] are used for comparison. Table 6 shows the accuracy for the specified number of training samples. It can be observed that for a large number of training samples LBP, CDA and CAE give high accuracy but it reduces rapidly as the number of training samples reduce. For 5 and 10 training samples our method achieves a significantly higher accuracy over these methods.

5.1.5. AR dataset

AR dataset presents two cases of occlusions, the lower portion of the face and the eye, by *sunglasses* and *scarf*, respectively. Following the protocol of [3] we randomly select 45 men and 45 women, and use their neutral face for training and tests are performed for expression and occlusions scenarios. Thus, there are 180 images in the training set and 560 images in the testing set. The results are compared with Eigenface, Fisherface, LBP, LGBP, GVLBP and its recent extensions E-GV-LBP-M and E-GV-LBP-P [3] and are shown in Table 7. It can be seen that for the expression test the proposed approach has the highest classification accuracy. For *sunglasses*, proposed method achieves more than 95% accuracy while the second best result achieves slightly more than 50% and rest of all have accuracy of less than 50%. The reason is clarified by observing the directional faces of the images with sunglasses in Fig. 11. The *sunglasses* occlude the eye region, however the algorithm is robust to these local intensity changes as long as the overall structure of the face image is maintained. However, if the overall structure of the face is disturbed, as is the case with the *scarf*, performance of the proposed approach degrades. Thus, for scarf it does not obtain the highest classification accuracy, however the accuracy is still much better than that of LBP and is not much below the best results.

5.1.6. FERET dataset

This dataset consists of a large number of subjects and it is one of the most widely used test protocols for evaluating the performance of face recognition algorithms. Here we follow the usual protocol [36,37] and conduct the experiments on the probe set Fb, Fc, Dup1 and Dup2. The images were resized to 60×60 pixels. The number of blocks for level $l=2$ is set to 5×5 and for level $l=3$ is set to 10×10 and level $l=0,1$ are set to the usual 1×1 and 2×2 blocks. The results for the four probe sets are shown in Table 8. Eigenface, Fisherface, LBP, LGBP, GVLBP, E-GV-LBP-M and E-GV-LBP-P [3] are used for comparison. It can be observed that the proposed algorithm has high accuracy for Fb, Fc and Dup1 probe tests, but for Dup2 it does not show high performance but still is much better than LBP and LGBP. It should be noted that this dataset does not have much illumination variations. On Extended Yale B and CMU-PIE that have extreme illumination variations, the proposed algorithm outperforms the other state-of-the-art approaches with a margin and on FERET it performs at par with them. So the proposed method is particularly robust to the illumination variations, which is still a major problem for the real time face recognition systems.

Table 7
AR face dataset.

Method	Expression	Sunglasses	Scarf
Eigenface	74.07	12.96	2.41
Fisherface	72.41	11.85	9.81
LBP	87.04	34.63	47.04
LGBP	86.11	37.59	82.59
GV-LBP	90.56	53.89	87.41
E-GV-LBP-M	90.93	42.77	82.78
E-GV-LBP-P	89.81	44.07	**86.67**
Ours	**92.78**	**95.50**	75.00

5.2. Noisy face Images

In this section we evaluate the performance of the proposed method on the noisy face images. We consider two different scenarios: (a) the additive Gaussian noise is assumed to be present in the face images (b) some pixels are missing from the face image which can be considered as a case of negative impulse noise.

5.2.1. Gaussian noise

Varying amount of Gaussian noise is added to ORL dataset with the standard deviations (sigma values) 4, 8 and 12. The noisy face image examples with different sigma are shown in Fig. 12. Gamma of ICI plays an important role in this case. It is set to 2.5, which allows considering higher scales, resulting in a robust estimate (as discussed in Section 4). Fig.13 shows the dependency of the classification accuracy on the sigma that varies from 0 to 12. It can be observed that even with sufficient amount of noise high classification accuracy is achieved. This can be easily explained by the fact that in noisy cases the ICI selects only the optimum scale which gives a robust estimate of the directional derivative.

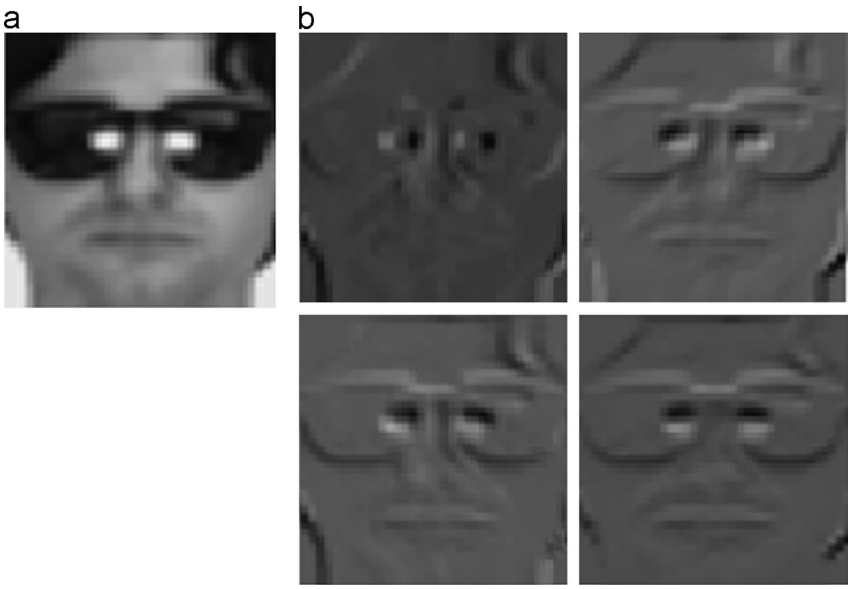


Fig. 11. (a) Face image with sunglasses and (b) ODFs of the image (a).

5.2.2. Impulse noise

The scenario of missing pixels is considered by adding the negative impulse noise to face images. The amount of noise is varied by measuring the percent of missing pixels that are varied from 0% to 8%. The image examples with missing pixels are shown in Fig. 12. The classification accuracy of the proposed method with respect to the percentage of missing pixels is shown in Fig. 14. The accuracies for the different corrupted images are 99.925%, 99.90%, 99.775%, 99.15%, and 98.65%, respectively. It can be seen that the accuracy remain above 99% till 6% of missing pixels and then drops quite gradually.

5.3. Discussion

Based on the observation of the test results discussed so far we can draw the following inferences:

1. Based on experiments performed on datasets with difficult illumination conditions (Extended Yale B and CMU PIE) the proposed algorithm achieves considerably high classification accuracy and outperforms other state-of-the-art approaches. It can be inferred that the proposed face representation technique is robust to illumination changes at both global and local levels. The local edges and boundaries of the face image provide discriminative features which are extracted by the directional derivatives under different illumination conditions. These discriminative directional features are nearly invariant to the illumination as they do not depend on the image intensity.
2. When comparing the dimensionality and computational complexity of Gabor and LPA based methods we can observe that the Gabor based method extracts features from 5 scales and 8 orientations while in LPA we use only 4 directions and 4 scales. These scales are further optimized for each direction. Thus, in Gabor based method 40 images are obtained while in our case the number of optimized directional faces is only 4. Extracting features from 40 images can pose computational problems and also a problem related to a dimensionality.
3. High accuracy of the proposed method can be attributed to the fact that features are extracted from the face images at different scales and levels. Directionality is considered by using directional filters, local neighborhood is considered to estimate the

Table 8
Classification accuracy for FERET dataset.

	Fb	Fc	Dup1	Dup2
Eigenface	78.91	9.79	33.66	11.97
Fisherface	87.78	47.42	44.32	20.09
LBP	97.00	79.00	66.00	65.00
LGBP	96.00	94.00	72.00	69.00
E-GV-LBP-M	**98.41**	**98.97**	81.99	**81.62**
E-GV-LBP-P	97.82	97.42	81.99	78.63
Ours	98.16	96.39	**83.90**	74.79

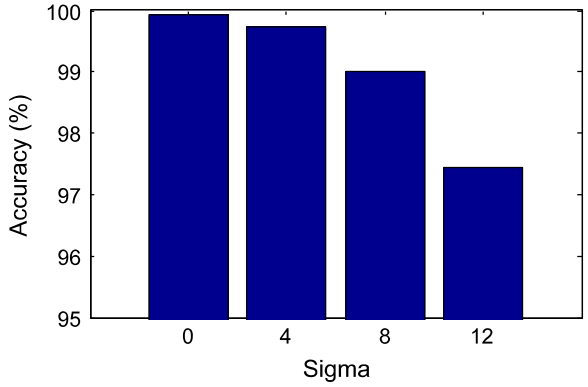


Fig. 13. The accuracy of ORL dataset when different amount of Gaussian noise is added to it.

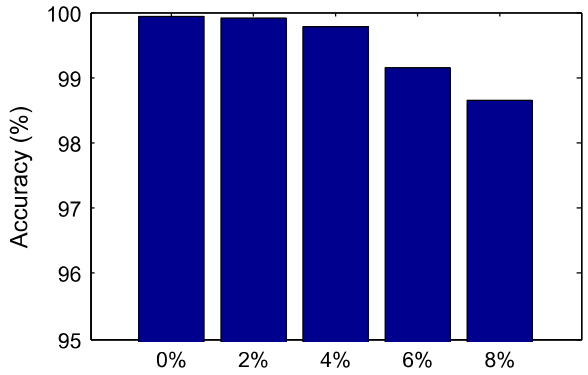


Fig. 14. Accuracy for ORL dataset when a certain percent of pixels are missing from the image.

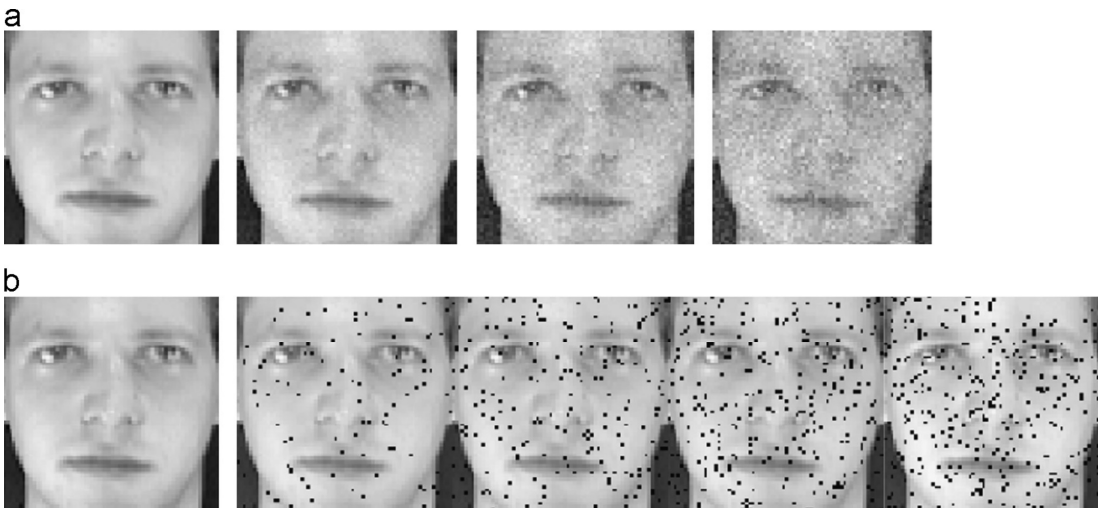


Fig. 12. (a) A face image and its corresponding Gaussian noisy images with $\sigma = 4$, 8 and 12. (b) A face image and next to it are noisy images with some missing pixels. The percentage of missing pixels is 2%, 4%, 6% and 8%.

derivatives by applying ICI in each direction, holistic information is extracted by applying modified LBP on the whole image and part based information is extracted at different levels by pyramid partitioning of the face image.

6. Conclusion

This paper proposes a novel method for automatic face recognition. In the proposed method we utilize LPA filter to extract directional information at multiple scales and further ICI is used to optimize these scales. Modified extension of LBP is used to allow the holistic and part based representation of the face image and at the same time to keep the dimensionality of the feature vector low. Capturing information at different levels from a face image, we obtain a robust feature vector which is invariant to illumination and pose changes. Experiments done on ORL, XM2VTS, Extended YaleB, CMU-PIE, AR and FERET datasets of face images show that the proposed method not only outperforms the benchmark methods like PCA and LDA but also a number of state-of-the-art methods for face recognition.

An important contribution of our work is that it introduces the directionality for recognition related task which has not been applied in such a manner, to the best of our knowledge. In future we plan to extend this method to other recognition related tasks.

Conflict of interest statement

None declared.

Appendix A. Design of window function

The LPA technique is used for nonparametric estimation using polynomial fit in a sliding window. The $w_{s,\theta}$ parameter in the LPA controls the scale and the direction of the sliding window. It determines the size of the neighborhood to be considered while estimating the value at a point. The directional window $w_{s,\theta}$ is obtained by rotating the two dimensional scalable window function w_s by angle θ around its center. We consider an unsymmetric uniform scalable window:

$$w_s(x) = \begin{cases} 1/(s-1) & \text{for } 0 \le x_1 < s, \quad |x_2| < 1/2 \\ 0 & \text{otherwise,} \end{cases} \tag{A1}$$

where $w_s(x)$ is on the two dimensional grid whose position is specified by $x = (x_1, x_2)$. From the above relation it can be observed that the window is unsymmetric along x_1 and symmetric along x_2.

References

[1] W.C. Zhang, S.G. Shan, W. Gao, H.M. Zhang, Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition, in: Proceedings of IEEE International Conference on Computer Vision, 2005, pp. 786–791.

[2] Z. Lei, S. Liao, R. He, M. Pietikäinen, S.Z. Li, Gabor volume based local binary pattern for face representation and recognition, in: Proceedings of IEEE International Conference on Automatic Face & Gesture Recognition, 2008, pp. 1–6.

[3] Z. Wenchao, S. Shiguang, C. Xilin, G. Wen, Local Gabor binary patterns based on Kullback–Leibler divergence for partially occluded face recognition, IEEE Signal Process. Lett. (2007) 875–878.

[4] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of Gabor phase patterns (HGPP): a novel object representation approach for face recognition, IEEE Trans. Image Process. (2007) 57–68.

[5] Z. Lei, S. Liao, M. Pietikainen, Z.S. Liu, Face recognition by exploring information jointly in space, scale and orientation, IEEE Trans. Image Process. (2010) 247.

[6] S. Xie, S. Shan, X. Chen, J. Chen, Fusing local patterns of Gabor magnitude and phase for face recognition, IEEE Trans. Image Process. (2010) 1349–1361.

[7] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognit. Neurosci. (1991) 71–86.

[8] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. (1997) 711–720.

[9] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. (2001) 228–233.

[10] Li-Fen Chen, Hong-Yuan, Mark Liao, Ming-Tat Ko, Ja-Chen Lin, Gwo-Jong Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition (2000) 1713–1726.

[11] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data—with application to face recognition, Pattern Recognition (2001) 2067–2070.

[12] J. Yang, J.Y. Yang, Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (3) (2003) 563–566.

[13] Gui-Fu Lu, Jian Zou, Yong Wang, Incremental complete LDA for face recognition, Pattern Recognition (2012) 2510–2521.

[14] F. Samaria, F. Fallside, Automated face identification using hidden Markov models, in: Proceedings of the International Conference on Advanced Mechatronics, 1993.

[15] L. Wiskott, J.-M. Fellous, N. Kruger, C.D. Von Malsburg, Face recognition by elastic bunch graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 775–779.

[16] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. (2002) 971–987.

[17] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2006) 2037–2041.

[18] T. Xiaoyang, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Trans. Image Process. (2010) 1635–1650.

[19] B. Zhang, Y. Gao, S. Zhao, J. Liu, Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor, IEEE Trans. Image Process. (2010) 533–544.

[20] A. Pentland, B..Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 84–91.

[21] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: Proceedings of SIGGRAPH, 1999, pp. 187–194.

[22] X. Li, W. Hu, Z. Zhang, H. Wang, Heat kernel based local binary pattern for face representation, IEEE Signal Process. Lett. (2009) 308–311.

[23] V. Katkovnik., K. Egiazarian, J. Astola, Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule, J. Math. Imaging Vision (2002) 223–235.

[24] P. Liang, S. Li, J. Qin, Multi-resolution local binary patterns for image classification, in: Proceedings of the Conference on Wavelet Analysis and Pattern Recognition, 2010, pp. 164–169.

[25] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: A Library for Support Vector Machines. Software available at: ⟨http://www.csie.ntu.edu.tw/∼cjlin/libsvm⟩, 2001.

[26] A. Foi, V. Katkovnik, K. Egiazarian, Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images, IEEE Trans. Image Process. (2007) 1395–1411.

[27] V. Katkovnik., K. Egiazarian, J. Astola, Frequency domain blind deconvolution in multiframe imaging using anisotropic spatially-adaptive denoising, in: EUSIPCO Proceedings of the 14th European Signal Processing Conference, 2006.

[28] V. Katkovnik, J. Astola., K. Egiazarian, Phase local approximation (PhaseLa) technique for phase unwrap from noisy data, IEEE Trans. Image Process. (2008) 833–846.

[29] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, K. Egiazarian, Spatially adaptive color filter array interpolation for noiseless and noisy data, Int. J. Imaging Syst. Technol. (2007) 105–122.

[30] R. Mehta, J. Yuan, K. Egiazarian, Local Polynomial Approximation-Local Binary Pattern (LPA-LBP) based face classification, in: Proceedings of SPIE-IS&T Electronic Imaging, Multimedia on Mobile Devices, 2011.

[31] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Matitre, XM2VTSDB: the extended M2VTS database, in: Proceedings of the Second International Conference on Audio- and Video-Based Biometric Person Authentication, 1999, pp. 965–966.

[32] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. (2001) 643–660.

[33] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. (2005) 684–698.

[34] T. Sim, S. Baker, M. Bsat, The CMU Pose, Illumination, and Expression (PIE) database, in: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 46–51.

[35] A. Martinez, R. Benavente, The AR face database, CVC, Technical Report 24, 1998.

[36] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, The FERET database and evaluation procedure for face recognition algorithms, Image Vision Comput. (1998) 295–306.

[37] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (2000) 1090–1104.

[38] T. Tan, H. Yan, Face recognition using weighted fractal neighbor distance, IEEE Trans. Syst. Man Cybern. C (2005) 576–582.

[39] Z. Li, D. Lin, X. Tang., Nonparametric discriminant analysis for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2009) 755–761.
[40] Y. Ma, S. Lao, E. Takikawa, M. Kawade, Discriminant analysis in correlation similarity measure space, in: Proceedings of International Conference on Machine Learning, 2007, pp. 577–584.
[41] Y. Fu, S. Yan, T.S. Huang, Correlation metric for generalized feature extraction, IEEE Trans. Pattern Anal. Mach. Intell. (2008) 2229–2235.
[42] V. Katkovnik, Multiresolution local polynomial regression: a new approach to pointwise spatial adaptation, Digital Signal Process. (2005) 73–116.
[43] S. Samaria, Andy Harter, Parameterization of stochastic model for human face identification, in: Proceedings of Application of Computer vision, 1994, pp. 138–142.

Rakesh mehta received M.S. degree from Tampere University of Technology in 2011. He is currently working in Institute of Signal Processing, Tampere University of Technology, Finland towards his Ph.D. His research interest includes face recognition and local texture descriptors.

Jirui Yuan received the B.S. degree in Communication Engineering in 2004 and M.S. degree in Communication & Information Systems in 2007 from Jilin University, China. Her research interests include video compression and joint source-channel video decoding. Currently, she is a Ph.D. student at the Institute of Signal Processing, Tampere University of Technology, Finland. Her recent work focused on face hallucination and recognition methods.

Karen Egiazarian (SM'96) was born in Yerevan, Armenia, in 1959. He received the M.Sc. degree in mathematics from Yerevan State University in 1981, the Ph.D. degree in physics and mathematics from Moscow State University, Moscow, Russia, in 1986, and the D.Tech. degree from the Tampere University of Technology (TUT), Tampere, Finland, in 1994. He was a Senior Researcher with the Department of Digital Signal Processing, Institute of Information Problems and Automation, National Academy of Sciences of Armenia. Since 1996, he has been an Assistant Professor with the Institute of Signal Processing, Tampere University of Technology, where he is currently a Professor, leading the Transforms and Spectral Methods Group. His research interests are in the areas of applied mathematics, signal processing, and digital logic.

Publication III: Multi-view Predictive Latent Space Learning. Jirui Yuan, Ke Gao, Pengfei Zhu, Karen Egiazarian. Pattern Recognition Letters, 2018.

Pattern Recognition Letters 000 (2018) 1–6

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Multi-view predictive latent space learning

Jirui Yuan[a], Ke Gao[b], Pengfei Zhu[b,*], Karen Egiazarian[a]

[a] Institute of Signal Processing, Tampere University of Technology, Tampere 999018, Finland
[b] School of Computer Science and Technology, Tianjin University, Tianjin 300072, China

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Multi-view learning
Predictive latent space learning
Unsupervised clustering
Unsupervised dimension reduction

ABSTRACT

In unsupervised circumstances, multi-view learning seeks a shared latent representation by taking the consensus and complementary principles into account. However, most existing multi-view unsupervised learning approaches do not explicitly lay stress on the predictability of the latent space. In this paper, we propose a novel multi-view predictive latent space learning (MVP) model and apply it to multi-view clustering and unsupervised dimension reduction. The latent space is forced to be predictive by maximizing the correlation between the latent space and feature space of each view. By learning a multi-view graph with adaptive view-weight learning, MVP effectively combines the complementary information from multi-view data. Experimental results on benchmark datasets show that MVP outperforms the state-of-the-art multi-view clustering and unsupervised dimension reduction algorithms.

1. Introduction

Multi-view learning becomes increasingly attractive due to the explosion of multi-view data, including samples represented by different feature descriptors, and objects from multiple sources, e.g., text, image, audio and video [24]. To exploit the complementary information from multiple views, numbers of multi-view learning methods have been developed and achieve superior performances in comparison with single-view learning. Analogous to other machine learning tasks, multi-view learning also suffers from two phenomena in the big data era, i.e., the lack of label information and high-dimensionality of the feature space [13]. The high expense of labelling data and the data explosion make most data unlabelled. Without label information, multi-view learning becomes more challenging in the unsupervised case [22].

Generally, it is intuitive to concatenate the multi-view data directly, and thus the traditional single-view solutions can be applied. Whereas, a simple concatenation will aggravate the curse of dimensionality, ignore the complementary nature and destroy distinct statistical properties of different views [6]. For unsupervised learning, one key challenge is how to discover the data structure by clustering. Multi-view clustering extends the clustering techniques in single-view learning (e.g., spectral clustering [2,9,10,16], linear regression [21] and matrix factorization [12]) to multi-view tasks. The common part of different views is usually modelled by cluster indicator matrix and diverse graph learning algorithms are developed. To alleviate the effect arising from the curse of dimensionality, dimension reduction algorithms are proposed to project the samples to a low-dimensional feature subspace. Thus, the time complexity and the computation burden are reduced while the generalization ability of the learning machines can be improved [8]. Traditional methods, like CCA [1,19] and PLS [18], can be used to cope with the two-view case. To exploit the correlations of multiple views simultaneously, researchers proposed many multi-view dimension reduction techniques [6,20,22,23,25]. Multi-view dimension reduction shares many similarities with multi-view clustering, and adopts similar techniques (e.g., matrix factorization and spectral analysis) to learn a projection matrix for each view.

For multi-view unsupervised learning, the key issue is to learn a latent representation for all views. In unsupervised learning, sample similarity relationships are often used in learning the latent space for both clustering and dimension reduction. For matrix factorization based methods, the latent space should well recover the raw feature space. However, almost all existing models ignore the predictive ability of the feature space, which is quite important for unsupervised learning. An undirected latent space Markov network was proposed to discover a predictive latent subspace representation shared by multiple views [3]. A latent space learning model was proposed to connect the feature space and label space for multi-label classification with many classes [11]. Nevertheless, they are specially designed for supervised learning.

In this paper, we propose a novel multi-view predictive latent space learning model (MVP) for multi-view unsupervised learning. MVP can be applied to both multi-view clustering and multi-

* Corresponding author.
E-mail address: zhupengfei@tju.edu.cn (P. Zhu).

https://doi.org/10.1016/j.patrec.2018.06.022
0167-8655/

view unsupervised dimension reduction. MVP learns a latent predictive representation by maximizing the correlation between the feature space of each view and the latent space. As the latent space is shared by all views, the consensus principle is fully used for multi-view learning. Considering the complementary nature, MVP learns a weighted graph to preserve the locality in multiple feature spaces jointly. Experimental results on datasets with multiple features show that MVP is superior to the state-of-the-art multi-view clustering and the unsupervised dimension reduction algorithms.

2. Related work

In this section, we will briefly review two typical learning tasks in multi-view unsupervised learning.

2.1. Multi-view clustering

Multi-view clustering algorithms have been developed to cluster data from multiple views simultaneously, by deriving a solution which uncovers the common latent structure shared by multiple views. Spectral clustering makes use of the spectrum of the similarity matrix of the data to discover the hidden data clusters [15]. Co-regularized multi-view spectral clustering (Co-regSC) is a multi-view spectral clustering framework by co-regularizing the clustering hypotheses [9]. Multi-modal spectral clustering (MMSC) considers each type of feature as a modal, and integrates such heterogeneous features by learning a commonly shared graph Laplacian matrix [2]. Multi-view non-negative matrix factorization (Multi-NMF) is a NMF-based multi-view clustering algorithm, and it formulates multi-view learning as a joint matrix factorization process [12]. Wang et al. [21] integrated all features of different views and used joint structured sparsity-inducing norms to learn a weight for each feature. Multi-view spectral clustering (MVSC) is a large-scale approach based on the bipartite graph to solve the massive data problem [10]. Auto-weighted multiple graph learning (AMGL) is a parameter-free multi-view model that can learn an optimal weight for each view automatically [16].

2.2. Multi-view unsupervised dimension reduction

Dimension reduction is one of the most important applications in unsupervised learning. It aims to map the high dimensional data into a low dimensional subspace, in which the similar samples in original space stay close to each other. Principle component analysis (PCA) is a typical single-view unsupervised dimension reduction method, which aims to find a subspace by maximizing the covariance of the input data points [8]. Multi-view dimension reduction have become a new topic in recent years due to the explosion of multi-view data. For unsupervised applications, canonical correlation analysis (CCA) is a classical multi-view method [19]. It learns a common feature space by maximizing the correlation between two projected spaces. Partial least square regression (PLS) also projects samples from two views to a common latent subspace, in which samples from one view are modelled as the regressor and samples from another view as the response [18]. CCA and PLS can only deal with two views, and ignore the correlation of multiple views. Xia et al. [22] proposed a distributed spectral embedding framework (DSE) that maps multiple views to a common linear subspace. Han et al. [6] developed a structured sparse multi-view embedding model (SSMVE) aiming to find a low-dimensional optimal consensus representation from multiple heterogeneous features. Zhao et al. [26] presented a graph-based co-training variant of locality preserving projection (LPP), named Co-LPP. Luo et al. [14] proposed a tensor CCA model, which naturally generalizes CCA to handle the data of an arbitrary number of (more than two) views. Wang et al. [20] introduced a multi-view sparsity preserving projection (MvSPP) method, which seeks a common low-dimensional subspace where multi-view sparse reconstruction weights are preserved as much as possible. Xie et al. [23] developed a multi-view exclusive unsupervised dimension reduction method (MEUDR) that considers the structured sparsity at both intra-view and inter-view levels. Zhang et al. [25] presented a multi-view dimension co-reduction model (MDcR) that explores the correlations among multiple views by a kernel method i.e. Hilber–Schmidt Independence Criterion.

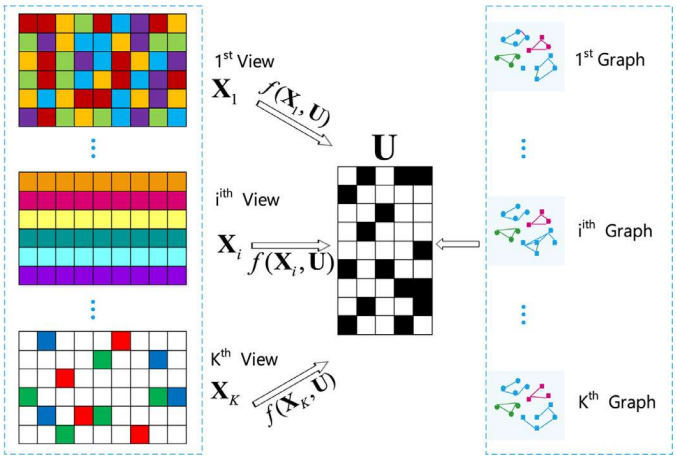


Fig. 1. The framework of multi-view predictive latent space learning (MVP).

In multi-view unsupervised learning, due to the lack of label information, the latent representation shared by multiple views, is usually expected to be discriminative and predictive. Whereas, the current matrix factorization and spectral analysis based methods do not emphasize the predictability of the latent space. In this paper, we will investigate the learning of a predictive latent space for multi-view unsupervised tasks.

3. Predictive latent space learning

In this section, we present the proposed multi-view predictive latent space learning.

3.1. Model

Let $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_K]$ be a set of multi-view data. $\mathbf{X}_i \in \mathbb{R}^{n \times d_i}$ is the data matrix of the ith view, where d_i is the feature dimension of the ith view and n is the number of samples. For multi-view learning, $\mathbf{X}_i$ can be different types of features, e.g., LBP, SIFT, GIST or Gabor. Additionally, $\mathbf{X}_i$ can be different kinds of modalities, e.g, text, image, video, etc.

As shown in Fig. 1, multi-view unsupervised learning aims to learn a latent representation for all views. There are two important principles in multi-view learning, i.e., the consensus principle and the complementary principle [24]. Whereas, because there is no class information, unsupervised learning in multi-view tasks becomes much challenging. Therefore, a latent representation with outstanding predictability is expected. Assume that $\mathbf{U} \in \mathbb{R}^{n \times h}$ is the latent space for all views, where h is the dimension of the latent space. The latent space is a low-dimensional representation of each view that can well characterize the discriminant structure embedded in multi-view high-dimensional data. Inspired by the definition of the predictability for the latent space in [11], we formulate the predictability of $\mathbf{U}$ by the correlation between the ith feature space and the latent space as

$$f(\mathbf{X}_i, \mathbf{U}) \tag{1}$$

Please cite this article as: J. Yuan et al., Multi-view predictive latent space learning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.06.022

where f is a function to measure the correlation between $\mathbf{X}_i$ and $\mathbf{U}$. The correlation $f(\mathbf{X}_i, \mathbf{U})$ should be maximized to enhance the predictability of the latent space $\mathbf{U}$. We use $\mathbf{u}$ to represent a column of $\mathbf{U}$, Therefore, the correlation between the feature space $\mathbf{X}_i$ and $\mathbf{u}$ can be represented by $f(\mathbf{X}_i, \mathbf{u})$.

$$f(\mathbf{X}_i, \mathbf{u}) = \frac{(\mathbf{X}_i\mathbf{v})^T\mathbf{u}}{\sqrt{(\mathbf{X}_i\mathbf{v})^T\mathbf{X}_i\mathbf{v}}\sqrt{\mathbf{u}^T\mathbf{u}}} \tag{2}$$

where $\mathbf{v}$ is a linear projection for $\mathbf{X}_i$. The orthogonal constraint is imposed on the latent space $\mathbf{U}$, i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. Then the follow optimization problem is introduced to maximize $f(\mathbf{X}_i, \mathbf{u})$.

$$\max_{\mathbf{u}} (\mathbf{X}_i\mathbf{v})^T\mathbf{u} \;\; s.t. \; (\mathbf{X}_i\mathbf{v})^T\mathbf{X}_i\mathbf{v} = 1 \tag{3}$$

When $\mathbf{u}$ is fixed, we use the method of Lagrange multipliers to calculate the optimal $\mathbf{v}$, denoted as $\hat{\mathbf{v}}$:

$$\hat{\mathbf{v}} = \frac{\left(\mathbf{X}_i{}^T\mathbf{X}_i\right)^{-1}\mathbf{X}_i{}^T\mathbf{u}}{\sqrt{\mathbf{u}^T\mathbf{X}_i(\mathbf{X}_i{}^T\mathbf{X}_i)\mathbf{X}_i{}^{-1}\mathbf{u}}} \tag{4}$$

Then the maximal $(\mathbf{X}_i\mathbf{v})^T\mathbf{u}$ can be derived as:

$$(\mathbf{X}_i\hat{\mathbf{v}})^T\mathbf{u} = \sqrt{\mathbf{u}^T\mathbf{\Lambda}_i\mathbf{u}} \tag{5}$$

where $\mathbf{\Lambda}_i = \mathbf{X}_i(\mathbf{X}_i^T\mathbf{X}_i)^{-1}\mathbf{X}_i^T$. Hence, to improve the predictability of the latent space, each column $\mathbf{u}$ of $\mathbf{U}$ is supposed to satisfy (Eq. (5)). The objective function for the total predictability of the latent space $\mathbf{U}$ in multi-view learning can be formulated as:

$$\max_{\mathbf{U}} \sum_{i=1}^{K} tr(\mathbf{U}^T\mathbf{\Lambda}_i\mathbf{U}) \;\; s.t. \;\; \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{6}$$

Eq. (6) takes the consensus principle into account by learning a common representation for all views. However, it does not consider the local geometry structure of the feature space in each view and the complementary information is not exploited as well. Hence, we introduced a weighted multi-graph regularization for the latent space. Then, we model manifold regularized latent space learning as:

$$\begin{aligned} &\max_{\mathbf{U},w} \sum_{i=1}^{K} tr(\mathbf{U}^T(\mathbf{\Delta}_i^* + \alpha w_i^r\mathbf{L}_i^*)\mathbf{U}) \\ &s.t. \sum_{i=1}^{K} w_i = 1, \; w_i \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I} \end{aligned} \tag{7}$$

where $\mathbf{L}_i^* = \mathbf{D}_i^{-1/2}\mathbf{W}_i\mathbf{D}_i^{-1/2}$ is the Laplacian matrix for the ith view, w_i is the weight for the ith view and $\alpha > 0$ is a tradeoff parameter. And we normalize $\mathbf{\Lambda}_i$ just like the Laplacian matrix $\mathbf{L}_i$ and denote it by $\mathbf{\Delta}_i^*$. In Eq. (7), we introduce a weight for each view, with a parameter $r > 0$. However, for the maximization problem, it is obvious that only the w_i of the maximum $tr(\mathbf{U}^T\mathbf{L}_i^*\mathbf{U})$ is close to 1, and others are 0. It means that only one view is selected by this method, which does not coincident with our objective on exploring the complementary property of multiple views. Hence, we transform the objective function into a minimization problem:

$$\begin{aligned} &\min_{\mathbf{U},w} \sum_{i=1}^{K} tr(\mathbf{U}^T(\mathbf{\Delta}_i + \alpha w_i^r\mathbf{L}_i)\mathbf{U}) \\ &s.t. \sum_{i=1}^{K} w_i = 1, \; w_i \geq 0, \mathbf{U}^T\mathbf{U} = \mathbf{I} \end{aligned} \tag{8}$$

where $\mathbf{\Delta}_i = \mathbf{I} - \mathbf{\Delta}_i^*$, $\mathbf{L}_i = \mathbf{I} - \mathbf{L}_i^*$, and $\mathbf{I}$ denotes an identity matrix.

In this paper, we adopt a parameter r to modulate the effect of the smoothness difference of graphs. The same phenomenon will occur when $r = 1$, that the effect of this difference is expanded. Only w_i of the minimum $tr(\mathbf{U}^T\mathbf{L}_i\mathbf{U})$ is close to 1, and other entries in w is 0. To address this problem, we set $r > 1$.

3.2. Optimization and algorithm

In this selection, we summarize the detailed optimization procedures of the model. We use an alteration minimization method to solve the optimization problem in Eq. (8). Then we update $\mathbf{U}$ and $\mathbf{w}$, respectively.

Subproblem U By fixing $\mathbf{w}$ and omitting the irrelevant items with respect to $\mathbf{U}$, the objective function can be written as:

$$\begin{aligned} &\min_{\mathbf{U}} tr\left(\mathbf{U}^T\sum_{i=1}^{K}(\mathbf{\Delta}_i + \alpha w_i^r\mathbf{L}_i)\mathbf{U}\right) \\ &s.t. \;\; \mathbf{U}^T\mathbf{U} = \mathbf{I} \end{aligned} \tag{9}$$

For the optimization problem in Eq. (9), we can decompose each column $\mathbf{u}$ of the matrix $\mathbf{U}$ into an optimization sub-problem. Each sub-problem should satisfy the following condition when we introduce a Lagrange multiplier to solve the problem with an equality constraint.

$$\sum_{i=1}^{K}(\mathbf{\Delta}_i + \alpha w_i^r\mathbf{L}_i)\mathbf{u} = \lambda\mathbf{u} \tag{10}$$

where λ is the introduced Lagrange multiplier for $\mathbf{u}$. Hence, we transform the optimization for $\mathbf{U}$ to a general eigenvalue problem.

Subproblem w When $\mathbf{U}$ is fixed, the optimization problem becomes only relevant to $\mathbf{w}$. The objective function degenerates into:

$$\begin{aligned} &\min_{w} tr\left(\mathbf{U}^T\sum_{i=1}^{K}(\mathbf{\Delta}_i + \alpha w_i^r\mathbf{L}_i)\mathbf{U}\right) \\ &s.t. \sum_{i=1}^{K} w_i = 1, \; w_i \geqslant 0 \end{aligned} \tag{11}$$

By using a Lagrange multiplier ξ to take the constraint $\sum_{i=1}^{K} w_i = 1$ into consideration, we get the Lagrange function as follows:

$$\mathcal{L}(w, \xi) = \left\{ tr(\mathbf{U}^T\textstyle\sum_{i=1}^{K}(\mathbf{\Delta}_i + \alpha w_i^r\mathbf{L}_i)\mathbf{U}) \right\} \left\{ -\xi\left(\textstyle\sum_{i=1}^{K} w_i - 1\right) \right\} \tag{12}$$

By setting the derivative of $\mathcal{L}(w, \xi)$ w.r.t. w_i and ξ to zero, we have:

$$\begin{cases} \frac{\partial\mathcal{L}(w,\xi)}{\partial w_i} = r\alpha w_i^{r-1}tr(\mathbf{U}^T\mathbf{L}_i\mathbf{U}) - \xi = 0 \\ \frac{\partial\mathcal{L}(w,\xi)}{\partial\xi} = \sum_{i=1}^{K} w_i - 1 = 0 \end{cases} \tag{13}$$

After Eq. (13) is solved, the updating formula for w_i can be obtained:

$$w_i = \frac{\left(1/tr(\mathbf{U}^T\mathbf{L}_i\mathbf{U})\right)^{1/(r-1)}}{\sum_{i=1}^{K}\left(1/tr(\mathbf{U}^T\mathbf{L}_i\mathbf{U})\right)^{1/(r-1)}} \tag{14}$$

The Laplacian matrix L_i is positive semi-definite, so we have $w_i \geq 0$ naturally. According to Eq. (14) we can find that r modulates the effect of the smoothness difference of graphs. We summarize the process of the MVP algorithm in Algorithm 1.

Time complexity We use an alternation maximization strategy for the proposed MVP. The main computation burden lies in the updating of the latent space $\mathbf{U}$ by Eq. (10). In each iteration, the time complexity of updating $\mathbf{U}$ is $O(n^3)$, where n is the dimension of samples. Let T be the iteration number of MVP. The time complexity of MVP is $O(Tn^3)$

Convergence analysis For the convergence of MVP, it can be easily proved that the optimization problem in Eq. (8) can converge to a local optimum on the basis of the convergence analysis in [22]. We empirically find that the proposed MVP method converges rapidly, as shown in Fig. 2.

Please cite this article as: J. Yuan et al., Multi-view predictive latent space learning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.06.022

Algorithm 1 The algorithm of MVP.

Require:

Data for K views $\mathbf{X}=[\mathbf{X}_1,\ldots,\mathbf{X}_i,\ldots,\mathbf{X}_K]\in\mathbb{R}^{n\times d}$.

1: Initialize $w_i=\frac{1}{m}$. Compute Laplacian matrix $\mathbf{L}_i$ and $\mathbf{\Delta}_i$ for each view. Calculate $\mathbf{L}=\sum_{i=1}^{m}(\alpha w_i^r\mathbf{L}_i)$.

2: **repeat**

3: Compute $\mathbf{U}$ by Eq. (10);

4: Update $\mathbf{w}$ by Eq. (14);

5: **until** Convergence criterion satisfied.

Ensure:

Latent space $\mathbf{U}\in\mathbb{R}^{n\times h}$

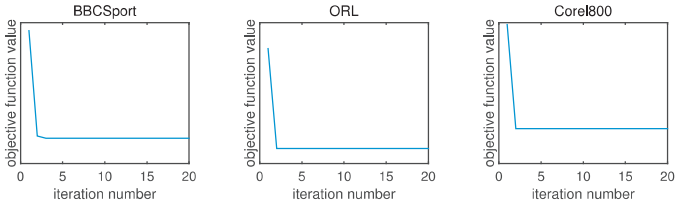


Fig. 2. The convergence curve of MVP for all the datasets.

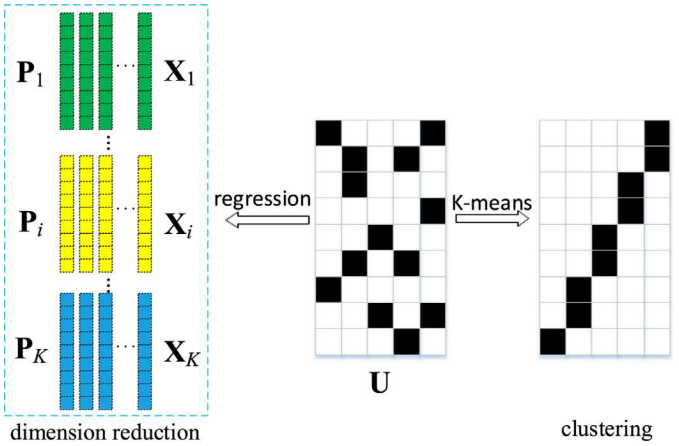


Fig. 3. Applications to multi-view clustering and unsupervised dimension reduction.

4. Applications to multi-view clustering and unsupervised dimension reduction

MVP learns a predictive latent representation for all views. Then we should consider how to use the latent space $\mathbf{U}$ for multi-view clustering and unsupervised dimension reduction.

MVP for clustering. We consider the latent space $\mathbf{U}$ as a new representation of multi-view data. Then we operate k-means algorithm on $\mathbf{U}$ to get the clustering labels, as shown in Fig. 3.

MVP for unsupervised dimension reduction. By learning a projection matrix $\mathbf{P}_i\in\mathbb{R}^{d_i\times h}$ that maps the samples from the high-dimensional feature space $\mathbf{X}_i$ to the low-dimensional latent space $\mathbf{U}$, we can reduce the dimensionality of the feature space for each view by using Eq. (15)

$$\hat{\mathbf{P}}_i=\arg\min\|\mathbf{X}_i\mathbf{P}_i-\mathbf{U}\|_F^2 \tag{15}$$

Eq. (15) has a closed-form solution $\hat{\mathbf{P}}_i=(\mathbf{X}_i^T\mathbf{X}_i)^{-1}\mathbf{X}_i^T\mathbf{U}$. To avoid trivial solution, a regularized solution can be used, i.e., $\hat{\mathbf{P}}_i=(\mathbf{X}_i^T\mathbf{X}_i+\varepsilon\mathbf{I})^{-1}\mathbf{X}_i^T\mathbf{U}$, where ε is a very small positive constant. Then we get a set of projection matrices $\{\mathbf{P}_1,\ldots,\mathbf{P}_i,\ldots,\mathbf{P}_K\}$ for all views. As shown in Fig. 3, the feature dimension of each view is reduced by the learned linear projections. Then, multi-view low-dimensional features are concatenated. Since the predictive latent space is used as the low-dimensional representation,it is expected that MVP can lead to better performances than the traditional models.

5. Experiments

In this section, we evaluate the proposed MVP on three benchmark datasets, and compare with the state-of-the-art multi-view unsupervised learning algorithms.

5.1. Experimental setup

Datasets. We use three datasets and the detailed descriptions about these datasets are given. 80% of samples are randomly selected for training and the rest for testing. Experiments are repeated for 10 times and the average result are reported.

- **BBCSport** dataset [4] consists of 544 sports news articles in five classes (athletics, cricket, football, rugby, tennis). It is a synthetic multi-view datasets. Each document is segmented and segments are randomly assigned to the two views (3183 and 3203 dimensions). At most one segment from each document is assigned to the same view [5].
- **ORL** face dataset [17] contains 400 different images of 40 distinct subjects, which are taken at different times, under varied lighting conditions and facial expressions. We resize the image into 64 × 64, and extract three types of features: intensity (4096 dimensions), LBP (3304 dimensions) and Gabor (6750 dimensions).
- **Corel800** dataset [7] contains 800 grayscale images of 10 individuals with 80 images per class. There are four types of feature, including LBP (59 dimensions) and GIST (512 dimensions) PHOG (680 dimensions), BOW feature (200 dimensions).

Evaluation metrics. We use one metric to evaluate the classification performance, and five evaluation metrics for clustering.

- **Classification accuracy** is a classification quality evaluation measure. It is the percentage of the total number of data points that are correctly classified.
- **Clustering accuracy** is a simple and transparent evaluation measure. It gives the percentage of the clustering result.
- **Clustering purity** is a general measure of clustering. To compute purity,[1] each cluster is assigned to the class which is the most frequent in the cluster. Then the accuracy of this assignment is measured by counting the number of correctly assigned data points.
- **NMI** normalizes the mutual information between the obtained clusters and the true clusters by the cluster entropies. NMI reaches its best value at 1 and worst at 0.
- **F-score** is a measure of test accuracy with ranges between 0 and 1. Higher values indicate closer match to the true clusters. It considers both precision and recall of the test stage.
- **ARI** is the abbreviation for adjusted rand index. It is a corrected-for-chance version of the rand index. The rand index is from 0 to 1 while the adjusted rand index is between −1 to 1.

5.2. Multi-view clustering

For unsupervised clustering, we compare the proposed MVP with the following methods:

[1] http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html.

Please cite this article as: J. Yuan et al., Multi-view predictive latent space learning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.06.022

Table 1
Clustering accuracy of multi-view clustering.

Dataset	BBCSport	ORL	Corel800
SC(1)	0.367(0.000)	0.475(0.004)	0.332(0.002)
SC(2)	0.369(0.000)	0.453(0.002)	0.285(0.002)
SC(3)	–	0.391(0.004)	0.312(0.003)
SC(4)	–	–	0.233(0.002)
Co-regSC	0.318(0.022)	0.482(0.020)	0.316(0.010)
MMSC	0.360(0.002)	0.160(0.008)	0.146(0.002)
AMGL	0.426(0.020)	0.694(0.022)	0.305(0.026)
MVP	**0.529(0.025)**	**0.714(0.037)**	**0.363(0.021)**

Table 2
NMI of multi-view clustering.

Dataset	BBCSport	ORL	Corel800
SC(1)	0.023(0.002)	0.692(0.004)	0.209(0.002)
SC(2)	0.020(0.000)	0.665(0.003)	0.184(0.002)
SC(3)	–	0.582(0.005)	0.234(0.003)
SC(4)	–	–	0.118(0.002)
Co-regSC	0.018(0.002)	0.677(0.006)	0.219(0.017)
MMSC	0.010(0.002)	0.347(0.008)	0.021(0.003)
AMGL	0.110(0.016)	0.827(0.022)	0.247(0.026)
MVP	**0.230(0.034)**	**0.867(0.020)**	**0.256(0.014)**

Table 3
Clustering purity of multi-view clustering.

Dataset	BBCSport	ORL	Corel800
SC(1)	0.373(0.001)	0.542(0.003)	0.343(0.002)
SC(2)	0.374(0.000)	0.516(0.003)	0.315(0.003)
SC(3)	–	0.473(0.005)	0.327(0.003)
SC(4)	–	–	0.250(0.002)
Co-regSC	0.367(0.002)	0.537(0.018)	0.339(0.010)
MMSC	0.361(0.002)	0.170(0.010)	0.151(0.003)
AMGL	0.443(0.017)	0.749(0.015)	0.353(0.021)
MVP	**0.538(0.026)**	**0.765(0.031)**	**0.375(0.021)**

- **BSV:** The performance of the most informative view is reported, i.e., one that achieves the best spectral clustering performance [15].
- **Co_regSC:** The co-regularized multi-view spectral clustering [9] looks for clusters that are consistent across the different views.
- **MMSC:** The multi-modal spectral clustering [2] considers each type of feature as one modal, and learns a commonly shared graph Laplacian matrix by unifying different modals.
- **AMGL:** Auto-weighted multiple graph learning [16] reformulates the standard spectral clustering learning model by learning a weight for each graph automatically without introducing an additive parameter.

There are two parameters for the proposed method, i.e., r and α in Eq. (8). We simply set r and α as 2 and 10 for all the datasets, respectively. For Co_regSC, we set the only parameter λ as 0.05, and the parameter r as 1 for MMSC according to the experimental setting in [2,9]. The AMGL is a parameter-free method. Tables 1–5 show the clustering accuracy, normalized mutual information (NMI), clustering purity, F-score, and ARI on the three datasets, respectively. We can see the proposed method MVP is superior to all the competing methods.

5.3. Multi-view unsupervised dimension reduction

For unsupervised dimension reduction, we compare our method with the following algorithms:

- **PCA:** We use the distributed PCA method [8], i.e., reducing the dimensions for every view by PCA, and then concatenating all the different views as a long vector.

Table 4
F-score of multi-view clustering.

Dataset	BBCSport	ORL	Corel800
SC(1)	0.388(0.000)	0.341(0.005)	0.205(0.001)
SC(2)	0.384(0.000)	0.290(0.004)	0.183(0.001)
SC(3)	–	0.176(0.005)	0.208(0.002)
SC(4)	–	–	0.147(0.000)
Co-regSC	0.325(0.010)	0.308(0.016)	0.207(0.030)
MMSC	0.385(0.006)	0.034(0.20)	0.099(0.022)
AMGL	0.381(0.008)	0.509(0.036)	0.208(0.008)
MVP	**0.440(0.012)**	**0.623(0.049)**	**0.246(0.010)**

Table 5
ARI of multi-view clustering.

Dataset	BBCSport	ORL	Corel800
SC(1)	0.008(0.000)	0.321(0.006)	0.116(0.001)
SC(2)	0.003(0.000)	0.267(0.004)	0.092(0.001)
SC(3)	–	0.143(0.005)	0.118(0.002)
SC(4)	–	–	0.053(0.000)
Co-regSC	0.005(0.029)	0.287(0.041)	0.118(0.002)
MMSC	0.006(0.023)	0.003(0.044)	-0.001(0.014)
AMGL	0.034(0.033)	0.495(0.021)	0.098(0.000)
MVP	**0.136(0.025)**	**0.615(0.051)**	**0.145(0.012)**

Table 6
Classification accuracy of unsupervised dimension reduction.

Dataset	BBCSport	ORL	Corel800
PCA	0.970(0.014)	0.975(0.017)	0.542(0.015)
NavieMDR	0.971(0.013)	0.976(0.016)	0.553(0.014)
MDcR	0.981(0.012)	0.973(0.016)	0.543(0.015)
DSE	0.963(0.009)	**0.981(0.012)**	0.580(0.010)
MVP	**0.984(0.008)**	**0.981(0.012)**	**0.586(0.014)**

Table 7
Clustering accuracy of unsupervised dimension reduction.

Dataset	BBCSport	ORL	Corel800
PCA	0.817(0.029)	0.804(0.029)	0.298(0.019)
NavieMDR	0.880(0.021)	0.761(0.021)	0.295(0.037)
MDcR	0.894(0.076)	0.812(0.076)	0.298(0.023)
DSE	0.887(0.041)	0.822(0.011)	0.369(0.031)
MVP	**0.896(0.003)**	**0.844(0.021)**	**0.370(0.025)**

- **DSE:** The distributed spectral embedding method [22] maps different views to a common linear subspace.
- **NaiveMDR:** It is a naive version of MDcR [25]. It reduces the dimension of each view without any correlation constraints between different views.
- **MDcR:** The Multi-view dimension co-reduction [25] explores the correlations within each view independently, and maximizes the dependence among different views.

The parameter setting of MVP is completely the same as that in multi-view clustering. For MDcR, we set the parameter λ as 10 according to Zhang et al. [25]. For other comparison algorithms, they are parameter-free. The reduced feature dimension is set as {10, 20, 30, 40, 50}. The best results of different feature dimensions are reported for all the methods. Table 6 shows the classification accuracy result. Tables 7 and 8 show the clustering accuracy and NMI, respectively. It clearly demonstrates that MVP improves the performance of multi-view concatenated features. Additionally, MVP outperforms the state-of-the-art unsupervised multi-view dimension reduction methods.

We also analyze the parameter sensitivity of MVP. As shown in Fig. 4, the performance of MVP is insensitive to α on three datasets for both multi-view clustering and dimension reduction. Therefore, we simply set α as 10.

Please cite this article as: J. Yuan et al., Multi-view predictive latent space learning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.06.022

Table 8
NMI of unsupervised dimension reduction.

Dataset	BBCSport	ORL	Corel800
PCA	0.809(0.034)	0.921(0.020)	0.232(0.017)
NavieMDR	0.770(0.065)	0.897(0.014)	0.237(0.005)
MDcR	0.808(0.036)	0.926(0.008)	0.233(0.016)
DSE	0.829(0.003)	0.918(0.005)	**0.303(0.019)**
MVP	**0.831(0.026)**	**0.936(0.008)**	0.276(0.006)

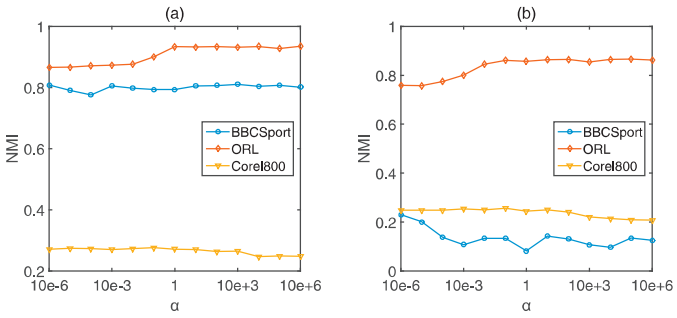


Fig. 4. Clustering performance with different values of α. (a) multi-view clustering. (b) multi-view unsupervised dimension reduction.

6. Conclusions

In this paper, we proposed a multi-view predictable latent space learning (MVP) model and applied MVP to multi-view clustering and unsupervised dimension reduction. Compared with the existing multi-view unsupervised learning models, MVP emphasizes the predictability of the latent representation shared by multiple views. The predictability of the latent space is modelled by the correlation between the feature space and the latent space. To combine the multi-view complementary information, a weighted multi-graph is learned when the multi-view correlations are maximized. Experiments are conducted on three datasets with multiple features for both clustering and dimension reduction. The results show that MVP achieves superior performances to the state-of-the-art algorithms for both applications.

Acknowledgement

This work was supported the (National Natural Science Foundation of China under Grants 61502332 and 61732011), Natural Science Foundation of Tianjin Under Grants 17JCZDJC30800.

References

[1] S. Akaho, A kernel method for canonical correlation analysis, arXiv preprint cs/0609071 (2006).
[2] X. Cai, F. Nie, H. Huang, F. Kamangar, Heterogeneous image feature integration via multi-modal spectral clustering, in: CVPR, IEEE, 2011, pp. 1977–1984.
[3] N. Chen, J. Zhu, E.P. Xing, Predictive subspace learning for multi-view data: a large margin approach, in: NIPS, 2010, pp. 361–369.
[4] D. Greene, P. Cunningham, Producing accurate interpretable clusters from high--dimensional data, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2005, pp. 486–494.
[5] D. Greene, P. Cunningham, A matrix factorization approach for integrating multiple data views, in: ECML-PKDD, Springer, 2009, pp. 423–438.
[6] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, J. Jiang, Sparse unsupervised dimensionality reduction for multiple view data, TCSVT 22 (10) (2012) 1485–1496.
[7] S.C.H. Hoi, W. Liu, M.R. Lyu, W.Y. Ma, Learning distance metrics with contextual constraints for image retrieval, in: CVPR, 2006, pp. 2072–2078.
[8] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2002.
[9] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: NIPS, 2011, pp. 1413–1421.
[10] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph., in: AAAI, 2015, pp. 2750–2756.
[11] Z. Lin, G. Ding, M. Hu, J. Wang, Multi-label classification via feature-aware implicit label space encoding., in: ICML, 2014, pp. 325–333.
[12] J. Liu, C. Wang, J. Gao, J. Han, Multi-view clustering via joint nonnegative matrix factorization, in: SDM, SIAM, 2013, pp. 252–260.
[13] B. Long, P.S. Yu, Z. Zhang, A general model for multiple view unsupervised learning, in: SDM, 2008, pp. 822–833.
[14] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, Y. Wen, Tensor canonical correlation analysis for multi-view dimension reduction, TKDE 27 (11) (2015) 3111–3124.
[15] A.Y. Ng, M.I. Jordan, Y. Weiss, et al., On spectral clustering: Analysis and an algorithm, in: NIPS, volume 14, 2001, pp. 849–856.
[16] F. Nie, J. Li, X. Li, et al., Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification, IJCAI, 2016.
[17] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on, IEEE, 1994, pp. 138–142.
[18] A. Sharma, D.W. Jacobs, Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch, in: CVPR, IEEE, 2011, pp. 593–600.
[19] B. Thompson, Canonical correlation analysis, Encycl. Stat. Behav. Sci. (2005).
[20] H. Wang, L. Feng, L. Yu, J. Zhang, Multi-view sparsity preserving projection for dimension reduction, Neurocomputing 216 (2016) 286–295.
[21] H. Wang, F. Nie, H. Huang, Multi-view clustering and feature learning via structured sparsity, in: ICML, 2013, pp. 352–360.
[22] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, TSMC-B 40 (6) (2010) 1438–1446.
[23] L. Xie, D. Tao, H. Wei, Multi-view exclusive unsupervised dimension reduction for video-based facial expression recognition, IJCAI, 2016.
[24] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv preprint (2013).
[25] C. Zhang, H. Fu, Q. Hu, P. Zhu, X. Cao, Flexible multi-view dimensionality co-reduction, TIP 26 (2) (2017) 648–659.
[26] X. Zhao, N. Evans, J.-L. Dugelay, Unsupervised multi-view dimensionality reduction with application to audio-visual speaker retrieval, in: Information Forensics and Security (WIFS), 2013 IEEE International Workshop on, IEEE, 2013, pp. 7–12.

Publication IV: Co-regularized Sparse Representation of Gaussians for Pattern Classification. Jirui Yuan, Hao Cheng, Karen Egiazarian. In submission to Pattern Recognition.

Publication V: Robust Deep Face Recognition with Label Noise. Yuan J, Ma W, Zhu P, Egiazarian K. International Conference on Neural Information Processing. Springer, Cham, 2017: 593-602.

Robust Deep Face Recognition with Label Noise

Jirui Yuan[1(✉)], Wenya Ma[2], Pengfei Zhu[2], and Karen Egiazarian[1]

[1] Tampere University of Technology, Tampere, Finland
jirui.yuan@student.tut.fi, karen.egiazarian@tut.fi
[2] School of Computer Science and Technology, Tianjin University, Tianjin, China
{wyma,zhupengfei}@tju.edu.cn

Abstract. In the last few years, rapid development of deep learning method has boosted the performance of face recognition systems. However, face recognition still suffers from a diverse variation of face images, especially for the problem of face identification. The high expense of labelling data makes it hard to get massive face data with accurate identification information. In real-world applications, the collected data are mixed with severe label noise, which significantly degrades the generalization ability of deep learning models. In this paper, to alleviate the impact of the label noise, we propose a robust deep face recognition (RDFR) method by automatic outlier removal. The noisy faces are automatically recognized and removed, which can boost the performance of the learned deep models. Experiments on large-scale face datasets LFW, CCFD, and COX show that RDFR can effectively remove the label noise and improve the face recognition performance.

Keywords: Deep learning · Noise removal · Face recognition

1 Introduction

Deep learning has achieved consistent breakthroughs in different tasks, including face recognition [1], scene understanding [2], and image caption [3]. The superior performance of deep learning owns to the representations of data with multiple levels of abstraction and massive labelled training data [4]. However, the lack of accurate label information makes it hard to learn a well-trained deep model with only a few labelled samples. For face recognition, despite the success of deep learning in face verification [5,6], it is hard to achieve satisfactory recognition accuracy without sufficient training data, especially when there are a large number of subjects in face identification. DeepFace uses a large-scale face dataset that consists of 4 millions face images of 4000 subjects [1]. FaceNet is learned on a much larger dataset with 200 millions of 8 millions subjects [5]. The large-scale face databases with accurate labels dramatically improve the performance of face recognition in that the deep learning models can be well trained.

How to acquire correctly labeled face dataset is one of the key challenges in constructing a successful face recognition system. One intuitive way is to manually collect and label the face images. The other way is a semiautomatic

D. Liu et al. (Eds.): ICONIP 2017, Part II, LNCS 10635, pp. 593–602, 2017.
https://doi.org/10.1007/978-3-319-70096-0_61

annotation by online image searching. The searching results contain massive label noise, which should be manually corrected. However, manual annotation suffers from high time consumption, labelling expense and inevitable labelling error [7]. Hence, there is a need to construct an effective face tagging method that can automatically remove noise, and allow collection of a large-scale face dataset with accurate identification information.

To deal with label noise, there are mainly three types of methods: noise-robust, noise-removal, and noise-tolerant. The first category of methods learn models that are robust to label noise. Manwani et al. proposed that when the loss functions are given, the learned model is claimed to be robust to noise if the misclassification probability is irrelevant to label noise [8]. Patrini et al. proposed to improve label noise robustness by loss factorization in weakly supervised learning [9]. Gao et al. divided the loss function into two parts: one irrelevant to noise and the other related with noise, by risk minimization [10]. The second type of methods consider that the noisy face images can be relabelled or directly discarded by a filter. These methods need to manually set a threshold for noise removal [11]. Wilson et al. reviewed the noise removal methods based on locality smoothness. Brodley et al. proposed to detect noisy samples by classification confidence scores [12]. The third type of methods model the noise distribution. Thus, the classification model and the noise model are directly separated. The most common noise modeling method is to estimate the noise distribution by the Bayesian methods.

For face recognition, noise removal aims to clean the noisy samples of each subject and then get a clean face dataset. Beside visual information, the side information can help to correct the label noise. Schroff et al. proposed to fuse visual and textual information to reorder the face images [13]. Li proposed to reorder the samples by incremental model learning using the searching results as the initialized rank [14]. Collins et al. used active learning to label a subset of face images helping noise removal. In real-world applications, the small-scale manually labelled face dataset and the side information maybe can be not reliable [15]. Hence, it is one of the most challenging issues to automatically detect noise samples in unsupervised setting and develop robust deep face recognition model.

In this paper, we propose a robust deep face recognition method by automatic label noise removal. A deep CNN model is firstly trained on a clean dataset with a small sample size. Deep features are extracted for a large-scale noisy face dataset by the pre-trained deep model. Then label noise is automatically removed by unsupervised one class learning (UOCL). Finally, a deep model is trained on the clean large-scale face dataset and tested on a validation set. This process is repeated until the recognition accuracy on the validation set does not increase. We use MS-Celeb-1M as the large-scale noisy dataset. Experiments on LFW, CCFD, and COX datasets shows that the proposed method can effectively alleviate the impact of label noise and improve the recognition performance of the learned deep models.

2 Robust Deep Learning

This section presents the proposed robust deep face recognition method.

2.1 Framework

The lack of data with accurate identification information blocks the improvement of the face recognition performance. Although it is easy to collect massive face images, the label noise may greatly degrade the performance of the recognition system. To make the best use of the large-scale noisy data, we propose a robust deep face recognition (RDFR) method by an automatic noisy removal. The framework is given in Fig. 1. Firstly, a deep model is trained on a clean dataset with a small sample size. The deep features are extracted for a large-scale noisy face dataset by the pre-trained deep model. Then, the noisy samples are removed by unsupervised one class learning (UOCL). This process is repeated to remove the noisy samples until the recognition rate on the validation set does not increase. RDFS aims to extract a clean subset from the large-scale noisy data to train a better deep model.

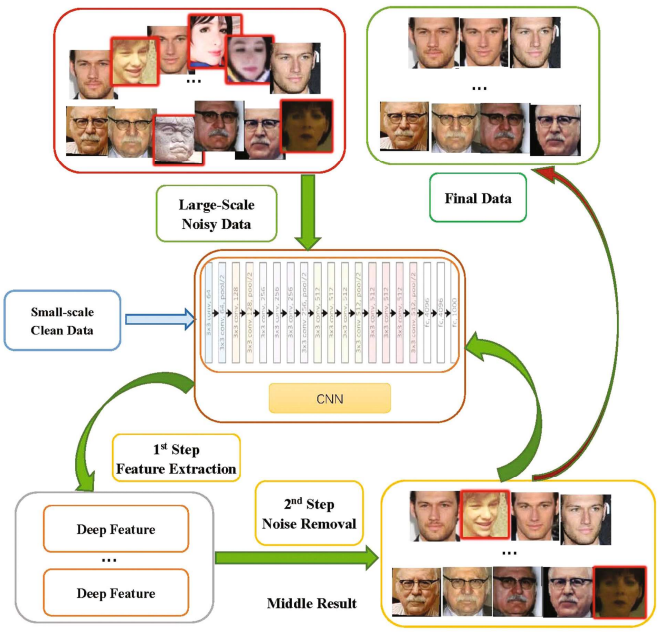


Fig. 1. The flowchart of robust deep face recognition via automatical label removal.

2.2 Unsupervised One Class Learning

In real-world applications, face images are more easily available and reliable compared to other information. Severe outliers should be removed from the large-scale dataset to make the visual information of face images well utilized. The common strategy to deal with a label noise is to transform outlier removal to an unsupervised one-class learning task. The representative methods are robust kernel density estimation (RKDE) [16] and sparse modeling for finding representative objects (SMRS) [17]. In this work, we introduce an efficient automatic noise removal method, namely, unsupervised one class learning (UOCL) [18]. UOCL is built upon two intuitive assumptions: (1) outliers originate from low-density samples, and (2) neighboring samples tend to have consistent classifications.

Given an unlabeled dataset $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ we aim to get a classification function $f : \mathbb{R}^d \mapsto \mathbb{R}$, which is similar to one class SVM. By leveraging a kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ that induces the Reproducing Kernel Hilbert Space (RKHS) the target classification function is in the following expression:

$$f(x) = \sum_{i=1}^{n} \kappa(\boldsymbol{x}, \boldsymbol{x}_i)\alpha_i, \tag{1}$$

where α_i is the expansion coefficient contributed by the functional base $\kappa(\cdot, \boldsymbol{x}_i)$. Let us introduce a soft label assignment $\mathcal{Y} = \{y_i \in \{c^+, c^-\}\}_{i=1}^n$, where c^+ is a positive value for positive samples and c^- is a negative value for outliers. Let $\boldsymbol{y} = [y_1, \cdots, y_n]^T$ be the vector representation of $\mathcal{Y}$.

Now we establish the UOCL model as minimizing the following objective:

$$\begin{aligned} \min_{f \in \mathcal{H}, \{y_i\}} \quad & \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i)^2 + \gamma_1 \|f\|_{\mathcal{M}}^2 - \frac{2\gamma_2}{n^+} \sum_{i, y_i > 0} f(\boldsymbol{x}_i) \\ \text{s.t.} \quad & y_i \in \{c^+, c^-\}, \forall i \in [1:n], \\ & 0 < n^+ = |\{i | y_i > 0\}| < n, \end{aligned} \tag{2}$$

where $\gamma_1, \gamma_2 > 0$ are two trade-off parameters controlling the model, $\|f\|_{\mathcal{M}}^2$ is the manifold regularization item.

2.3 Deep Model

For label noise removal, we use VIPLFaceNet and in the stage of face recognition, we use Resnet-VIPL. VIPLFaceNet contains 7 convolution layers and 3 full connected layers. Resnet-VIPL is modified from the classic Resnet [19], and consists of 82 convolution layers and 2 full connected layers. Compared with Resnet-101, Resnet-VIPL greatly reduces the computation burden while keeps the performance.

3 Experiments

Experiments are conducted on large-scale face databases to evaluate the performance of the proposed method (Fig. 2).

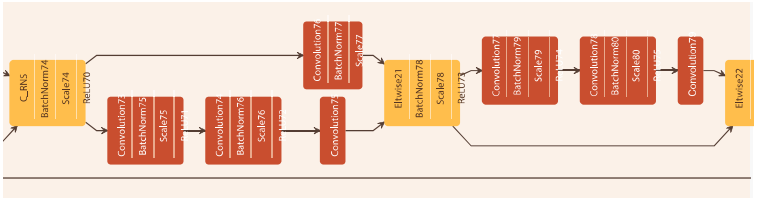


Fig. 2. A part of Resnet-VIPL.

3.1 Datasets

We use a large-scale noisy face dataset MS-Celeb-1M for training. The performance is evaluated on three datasets, including LFW, CCFD and COX.

MS-Celeb-1M is a large-scale noisy dataset from Microsoft [20]. MS dataset has 8,456,240 real-world facial images of 99,891 identities. It is a large-scale dataset that contains large variations in age, pose and so on. There are severe label noises, which may degrade the performance of deep models.

CCFD (Chinese Celebrity Face Dataset) is a large-scale real-world face dataset collected by VIPL. This dataset consists of 263,696 images of 1,001 subjects, with two subsets for training and testing. The training set contains 171,792 images of 701 subjects and the testing set contains 91,904 images of 301 subjects. Facial images in CCFD are collected from the internet and have large variations in age, expression, light, occlusion and pose.

LFW (Labeled Faces in the Wild) is a classic face dataset that consists of 13,233 images of 5,749 identities [21].

CASIA-WebFace is a public face dataset that consists of 494,414 images of 10,575 subjects [22].

COX consists of the gallery set and probe set. The gallery set contains 20,312 face images of 20,312 subjects. The images in the gallery set are the face images of the Chinese identity card. The probe set contains 1,102 test images, which are collected in the wild.

The comparison of different large-scale face datasets are illustrated in Table 1. The test protocols of the three datasets are different.

CCFD: The test set of CCFD contains 91,904 face images of 301 subjects. The test set is divided into the target set and the query set. The verification rate under different false acceptance rate is used to evaluate the recognition performance. Here, the verification rate when FAR is 0.1 is reported.

COX: The ROC curse is used to evaluate the performance.

LFW: The average face verification rate of ten folds are used. There are 300 positive pairs and 300 negative samples per fold.

3.2 Experimental Settings

Face preprocessing. The face images of differen datasets are all resized to 256 $\times$ 256. Deep features are extracted for label noise removal and

Table 1. The comparison of large-scale face datasets

Datasets	Subject	Image	Property
LFW	5,749	13,233	public(clean)
WDRef	2,995	99,773	private(clean)
CelebFace	10,177	202,599	public(clean)
MSRA-CFW	1,583	202,792	public(clean)
CCFD	1,001	270,706	private(clean)
CASIA-WebFace	10,575	494,414	public(clean)
SFC	4,030	4,400,000	private(clean)
MS-Celeb-1M	99,891	8,456,240	public(noise)
Google	8,000,000	200,000,000	private(clean)

face recognition. The deep feature dimension for noise removal is 2,048 and the dimension for face recognition is 1,024.

Parameter setting. The platform of our experiments is Caffe. SGD is utilized to train the VIPLFaceNet and Resnet-VIPL. For VIPLFaceNet, we set the base_lr as 0.06, mini-batch size as 128, iter_size as 1, total iteration in pre-train process as 120,000, momentum as 0.9, and weight-decay as 0.0002. The learning rate is decreased according to the polynomial policy with *gamma* value equals to 0.5. For Resnet-VIPL, we set the base_lr as 0.04, mini-batch size as 32, iter_size as 4, total iteration in pre-train process as 300,000, momentum as 0.9, and weight-decay as 0.0002. For UOCL, we use Gaussian kernel $k(x, y) = exp(-\|x - y\|^2) \ / \ 2\sigma^2)$, where $\sigma = \sum_{i,j=1}^{n} \|x_i - y_i\|^2 \ / \ n^2$ (Fig. 3).

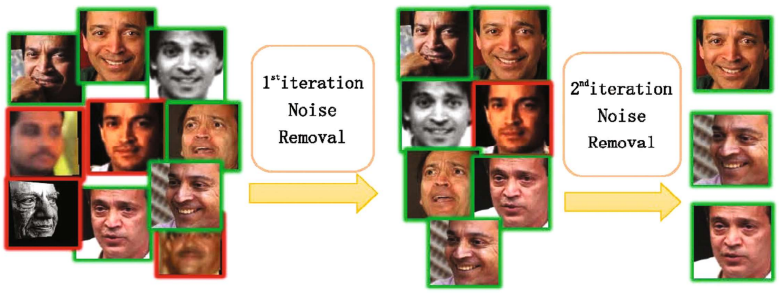


Fig. 3. The process of label removal on MS-Celeb-1M database. Red bounding box represents the correctly labelled samples while green bounding box represents noisy samples. The face images of one person is taken as example. (Color figure online)

3.3 Experimental Analysis

We use CASIA-WebFace as the clean dataset with small sample size to train a CNN model. Noisy samples are iteratively removed from MS-Celeb-1M. We compare the recognition rate of the raw noisy dataset and the clean dataset after noise removal. Figure 4 shows that by automatical noise removal, the noisy

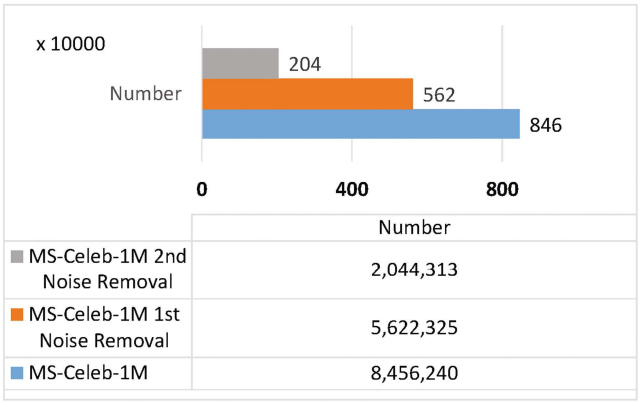


Fig. 4. The number of face samples in MS-Celeb-1M before and after noise removal.

Table 2. The face verification rate on LFW dataset

Method	Training dataset	Accuracy
Resnet-VIPL	MS-Celeb-1M	99.25%
Resnet-VIPL	MN_01	99.40%
Resnet-VIPL	MN_02	99.25%
DeepFace	SFC	97.35%
WSTFusion	WSTFusion	98.73%
VGGFace	VGGFace	98.95%
DeepID2+	DeepID2+	99.47%
FaceNet	Google	99.63%

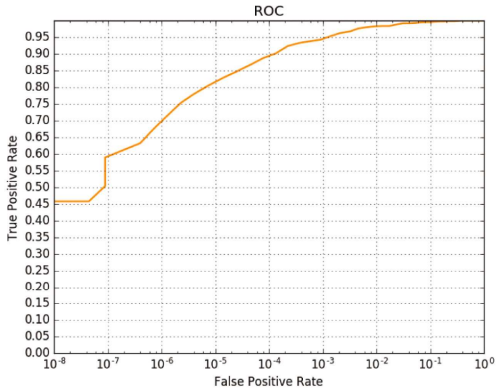
Table 3. The face recognition rate on CCFD dataset

Method	Training dataset	Finetune	Accuracy
Resnet-VIPL	MS-Celeb-1M	No	58.10%
Resnet-VIPL	MN_01	No	64.72%
Resnet-VIPL	MN_02	No	61.19%
Resnet-VIPL	MS-Celeb-1M	Yes	65.04%
Resnet-VIPL	MN_01	Yes	70.66%
Resnet-VIPL	MN_02	Yes	68.41%

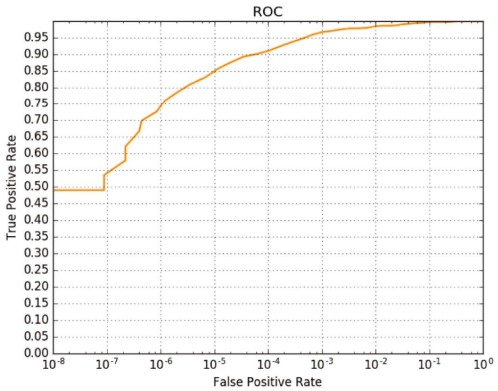
samples are partially removed after then first iteration. Then in the second iteration, all noisy samples are removed together with some clean samples. Hence, noise removal may discard also many clean face samples.

Figure 4 shows the number of samples left in MS-Celeb-1M. After the first iteration, about three million face images are removed while during the second iteration, the other three million samples are removed. The number of removed samples shows that during the iterations, we should carefully use the noise removal algorithm.

Table 2 shows the face verification rate on LFW dataset. MN_01 and MN_02 represent the results of the 1^{st} and the 2^{nd} noise removal. The results show that compared with the raw noisy data, the verification rate is improved by 0.25% after the 1^{st} noise removal. Compared with DeepFace, VGGFace and DeepID2+, the performance of the proposed method is superior or comparable.



(a) The ROC curve when MS-Celeb-1M is used for training



(b) The ROC curve when the cleaned MS-Celeb-1M is used for training

Fig. 5. The comparison of ROC curve on COX dataset

FaceNet achieves 99.63% in that it uses 200 million face images to train the deep model. After the second iteration, the recognition rate is the same as the raw noisy data. However, the number of training samples is only a quarter of the raw data. Hence, the time consumption and storage burden is greatly reduced.

Table 3 shows the recognition rate on CCFD dataset. Note, that the face images in MS-Celeb-1M are all collected from European and American while CCFD contains only the face images of Chinese Celebrities. To reduce the gap across different ethnic groups, we finetune the parameters on the training set of CCFD to improve the recognition performance. From the result, we can see that similar to LFW, the model trained on MN_01 is much better than on MS-Celeb-1M. Compared with the result without finetuning, the recognition rate is much improved. Note that after the second noise removal, the rate slightly decreases, since too many clean data have been removed together with the noisy face images.

Figure 5 shows results on COX dataset. The ROC curves before and after label noise removal clearly reflect the effectiveness of the proposed method.

4 Conclusions and Future Work

In this paper, we proposed a robust deep face recognition method by automatical noise removal. Because of the parameter explosion in deep learning techniques, a large-scale face dataset with correct label information is badly needed to train an accurate deep learning model. Unsupervised one-class learning is used to remove the massive noisy face images. Experiments on large-scale face datasets in the wild validate the effectiveness of the proposed method. In the future, we will focus on end-to-end robust deep face recognition model.

Acknowledgements. This work was supported by the National Program on Key Basic Research Project under Grant 2013CB329304, the National Natural Science Foundation of China under Grants 61502332, 61432011, 61222210.

References

1. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708. IEEE (2014)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
3. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4651–4659 (2016)
4. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
5. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)

6. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: Face recognition with very deep neural networks (2015). arXiv preprint: arXiv:1502.00873
7. Ariz, M., Bengoechea, J.J., Villanueva, A., Cabeza, R.: A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. In: Computer Vision and Image Understanding, vol. 148, pp. 201–210 (2016)
8. Manwani, N., Sastry, P.S.: Noise tolerance under risk minimization. IEEE Trans. Cybern. **43**(3), 1146 (2011)
9. Patrini, G., Nielsen, F., Nock, R., Carioni, M.: Loss factorization, weakly supervised learning and label noise robustness. In: International Conference on International Conference on Machine Learning, pp. 708–717 (2016)
10. Gao, W., Wang, L., Li, Y.F., Zhou, Z.H.: Risk minimization in the presence of label noise. In: AAAI, pp. 1575–1581 (2016)
11. Zhang, J., Sheng, V.S., Li, T., Wu, X.: Improving crowdsourced label quality using noise correction. IEEE Trans. Neural Netw. Learn. Syst. (2017)
12. Brodley, C.E., Friedl, M.A.: Identifying and eliminating mislabeled training instances. In: Thirteenth National Conference on Artificial Intelligence, pp. 799–805 (1996)
13. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. IEEE Trans. Pattern Anal. Mach. Intell. **33**(4), 754–766 (2011)
14. Li, L.J., Fei-Fei, L.: Optimol: automatic online picture collection via incremental model learning. Int. J. Comput. Vis. **88**(2), 147–168 (2010)
15. Collins, B., Deng, J., Li, K., Fei-Fei, L.: Towards scalable dataset construction: an active learning approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 86–98. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88682-2_8
16. Kim, J., Scott, C.D.: Robust kernel density estimation. J. Mach. Learn. Res. **13**, 2529–2565 (2012)
17. Elhamifar, E., Sapiro, G., Vidal, R.: See all by looking at a few: sparse modeling for finding representative objects. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1600–1607. IEEE (2012)
18. Liu, W., Hua, G., Smith, J.R.: Unsupervised one-class learning for automatic outlier removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3826–3833 (2014)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). arXiv preprint: arXiv:1512.03385
20. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part III. LNCS, vol. 9907, pp. 87–102. Springer, Cham (2016). doi:10.1007/978-3-319-46487-9_6
21. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report 07-49, University of Massachusetts, Amherst, October 2007
22. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch (2014). arXiv preprint: arXiv:1411.7923

Publication VI: Multi-task Deep Face Recognition. Yuan J, Ma W, Zhu P, Egiazarian K. Chinese Conference on Biometric Recognition. Springer, Cham, 2017: 183-190.

Multi-task Deep Face Recognition

Jirui Yuan[1(✉)], Wenya Ma[2], Pengfei Zhu[2], and Karen Egiazarian[1]

[1] Tampere University of Technology, Tampere, Finland
jirui.yuan@student.tut.fi, karen.egiazarian@tut.fi
[2] School of Computer Science and Technology, Tianjin University, Tianjin, China
{wyma,zhupengfei}@tju.edu.cn

Abstract. In recent years, deep learning has become one of the most representative and effective techniques in face recognition. Due to the high expense of labelling data, it is costly to collect a large-scale face dataset with accurate label information. For the tasks without sufficient data, deep models cannot be well trained. Generally, parameters of deep models are usually initialized with a pre-trained model, and then fine-tuned on a small dataset of specific task. However, by straightforward fine-tuning, the final model usually does not generalize well. In this paper, we propose a multi-task deep learning (MTDL) method for face recognition. The superiority of the proposed multi-task method is demonstrated by experiments on LFW and CCFD.

Keywords: Deep learning · Multi-task · Face recognition · Convolution neural network

1 Introduction

Deep learning has achieved great performances in face recognition because it can capture the face variations by learning hierarchical high-level representation. The recognition accuracy has exceeded 99.00% on the challenging benchmark LFW [1] dataset such as DeepID3 [2] achieves 99.53% and FaceNet [3] achieves 99.63%. A few latest results that break the record have been continuously reported recently. Deep convolutional neural network has become the most representative and effective technology for face recognition in the wild.

In many computer vision tasks, we only have a dataset with small sample size. Whereas, a deep learning model cannot be well trained without sufficient training data because of parameters explosion in deep neural networks. One possible solution is to train a model directly on the dataset with small sample size. This approach is simple enough but it is difficult for the model to obtain a satisfactory result. The better solution is to utilize the large-scale dataset to train a deep model, and the model can handle a specific task called task A. We can utilize the pre-trained model to fulfil task B which is a task related with task A by fine-tuning. Fine-tuning methods can utilize the relationship between the two datasets, but sometimes they can not avoid overfitting.

J. Zhou et al. (Eds.): CCBR 2017, LNCS 10568, pp. 183–190, 2017.
https://doi.org/10.1007/978-3-319-69923-3_20

Multi-task Learning (MTL) is an inductive transfer mechanism whose principle goal is to improve generalization performance. Generally, it jointly learns the parameters of multiple related tasks simultaneously by pursuing a shared presentation [4]. MTL works because it effectively increases the sample size by implicit data augmentation [5]. Additionally, by learning common representation for multiple tasks, the effect of noise in each task can be compressed. By capturing the similarity among different tasks, MTL can also avoid over-fitting on a specific task to some extent. MTL has already been used in multiple tasks including face recognition network DeepId2 [6], object detection network Faster R-CNN [7], fine-grained vehicle classification network [8], facial landmarks detection and attributes classification network TCDCN [5].

Motivated by multi-task learning, we propose a multi-task deep learning method for face recognition by using multiple face datasets. We consider learning on multiple datasets as a multi-task learning problem. First, we pre-train a model on a large-scale dataset and consider it as an initial task. Then for other specific tasks, we initialize parameters with the pre-trained model, and then fine-tune on the multiple datasets that consist of the large-scale dataset and the limited dataset of the specific tasks. The hidden layers are shared by all datasets while task-specific output layers are leaned for each dataset. This method makes full use of both the large-scale dataset with small sample size. In this way, the final model can get better performance than straightforward fine-tuning. Furthermore, the accuracy on the initial task can be maintained. Meanwhile, the pre-trained model can be migrated conveniently to other tasks with limited data and reduce time consumption of model training. Experiments on LFW and CCFD validate the effectiveness of the proposed multi-task deep learning model for face recognition.

The rest of this paper is organized as follows: Sect. 2 presents the multi-task deep learning framework. Section 3 introduces multi-task deep face recognition with different face datasets. Section 4 conducts experiments. Section 5 comes to the conclusion.

2 Multi-task Learning Model

In this section, we introduce the multi-task deep learning model.

2.1 Task Loss

Figure 1 shows the structure of the multi-task deep learning method. For face recognition, we train a deep model by using multiple face datasets. Here, each dataset is considered as an individual task. Assume that there are two datasets, i.e., one large-scale dataset and one limited dataset with small sample size. An accurate model can be pre-trained on the large-scale dataset. To utilize the relation between two datasets and avoid overfitting, both datasets are combined as the input layer and share the common representation, i.e., the hidden layers. Finally the data are split and dataset-specific fully connected layers are designed.

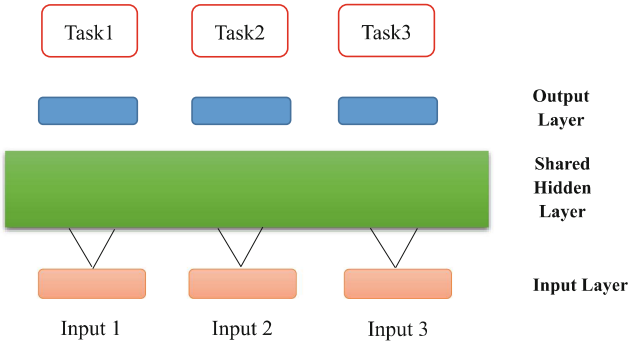


Fig. 1. Deep network for multi-task deep learning. Different inputs mean the different tasks of data. Task 1, Task 2 and Task 3 represent the loss function for each task. The proposed multi-task deep learning model learns shared hidden layers while tasks-specific output layers.

Assume that there are two related tasks that they are all classification tasks. Task A has C_A series and task B has C_B series. In Caffe [9], for each mini-batch, the loss function of task A is:

$$L_A = \frac{-1}{N_A} \sum_{n=1}^{N_A} \log(\hat{p}_{nl_n}), \quad l_n \in [0, 1, \ldots, C_A - 1], \tag{1}$$

where N_A is the images number in one mini-batch and $\hat{p}_{nl_n} = \frac{e^{x_n l_n}}{\sum_{c=0}^{C_A - 1} e^{x_n c}}$. Thus, the loss of two tasks is given as follow:

$$L_A = \frac{-1}{N_A} \sum_{n=1}^{N_A} \log(\frac{e^{x_n l_n}}{\sum_{c=0}^{C_A - 1} e^{x_n c}}), \quad l_n \in [0, 1, \ldots, C_A - 1], \tag{2}$$

$$L_B = \frac{-1}{N_B} \sum_{n=1}^{N_B} \log(\frac{e^{x_n l_n}}{\sum_{c=0}^{C_B - 1} e^{x_n c}}), \quad l_n \in [0, 1, \ldots, C_B - 1], \tag{3}$$

2.2 Back Propagation

As shown in Fig. 1, each task calculates its own loss. Whereas, during back propagation, all gradients will be added to update the parameters of the deep model. When learning a task, multi-task deep learning method can obtain the knowledge of other tasks with shared representation. Multiple tasks in parallel training share features learned from other different tasks, which is the main idea of multi-task learning. By adding all gradients to execute the back propagation, it makes sure that parameters in the deep convolutional neural network update with the right trend for model training of different tasks.

3 Multi-task Deep Learning for Face Recognition

In this section, we firstly introduce the basis of multi-task deep learning method based on multiple datasets and then introduce how to utilize multi-task deep learning method to train models. Figure 2 shows the overview of our approach.

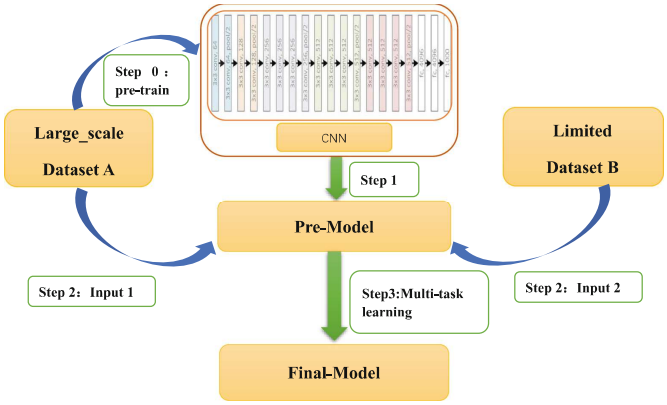


Fig. 2. Overview of our approach.

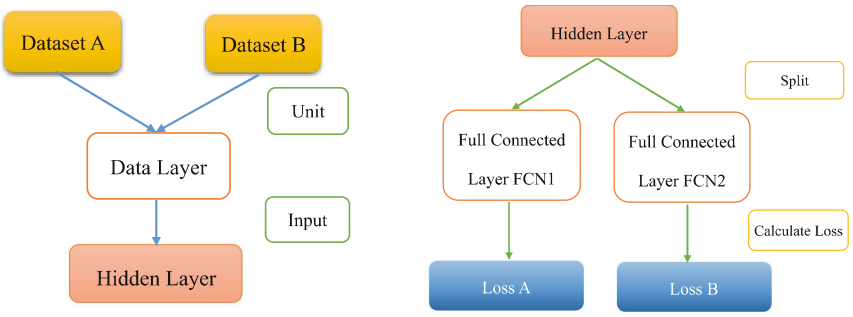
3.1 Pre-train Model

Our approach is based on fine-tuning method. In our approach, there is a large-scale dataset called D_A corresponding to task T_A. Firstly, we need find or construct a deep convolution neural network to train a deep model for T_A with D_A. This model has excellent accuracy on the T_A and then it will be used as pre-trained model to fine-tune with multiple dataset in the next steps. The model is the basis of our approach.

3.2 Multi-task Deep Learning for Multiple Datasets

In our approach, after pre-trained model was generated, it is time to carry on multi-task deep learning for multiple datasets. First, there are two datasets D_A for task T_A, D_B for task T_B. We need unit them as the input data used for model training. We unit multiple data in axis n as shown in Fig. 3. Then the combined data is used to train our convolution neural network. Note that we don not unit labels of multiple datasets.

The united data is splitted before classification. Here, data is splitted after the penultimate full connected layer. Generally, the penultimate fully connected layer carry on feature representation and the last fully connected layer do the classification work. Then every task calculate their own loss. The structure of



(a) Data concat. (b) Data split.

Fig. 3. Data concat and data split in our approach.

data split is shown in Fig. 3. In order to be consistent, data split at the axis of n. We only operate on the sample space and their labels is fixed.

In our approach, multiple tasks should be relevant. Multiple tasks share the convolution layers and some fully connected layers including feature representation layer in our next experiments. But in other real-world tasks, maybe we can only share convolution layers while learning task-specific fully-connected layers like deep relationship networks [10].

4 Experiments

In this section, we evaluate our approach on the real-world datasets. We conduct experiments to examine the multi-task deep learning method when we have a large-scale face dataset and a limited dataset. In our experiments, we use MS dataset to train a model and then utilize MS dataset and CCFD dataset for multi-task learning. In next subsections, we introduce datasets, settings and conclusions of our experiments.

4.1 Datasets

We investigate our approach on three datasets: MS dataset, CCFD dataset and LFW dataset.

MS: MS dataset is a subset of large-scale noisy dataset of real-world called MS-Celeb-1M [11]. After artificially marked, MS dataset is clean enough. MS dataset has 3,095,536 real-world facial images of 41,857 identities. It is a large-scale dataset that contains large variations in age, pose and so on.

CCFD: CCFD (Chinese Celebrity Face Dataset) is a large-scale real-world face dataset collected by VIPL. This dataset consists of 263,696 images of 1,001 subjects, with two subsets for training and testing. The training set contains 171,792 images of 701 subjects and the testing set contains 91,904 images of 301

subjects. Facial images in CCFD are collected from the internet and have large variations in age, expression, light, occlusion and pose.

In our approach, in order to ensure the consistency of the experiments, all the face images are normalized to 256×256 and five facial points are aligned. In addition, for face recognition, the detection tasks of European & American and Chinese are two different tasks because of the difference between human race. The faces in MS dataset are almost from European and American, and the identities in CCFD dataset are all Chinese. The two datasets meet the demand of our assumption of our multi-task deep learning method.

4.2 CNN and Settings

In our experiments, we use Resnet-VIPL as convolution neural network to train deep model. Resnet-VIPL evolves from Resnet [12] and it consists of 82 convolutional layers and 2 fully connected layers. It has less calculations compared to Resnet-101 but also has a satisfactory accuracy in experiments.

The platform of our experiments is Caffe. SGD is utilized to train the Resnet-VIPL. In all the experiments, we set The learning rate is decreased according to the polynomial policy with *gamma* value equals to 0.5. The base learning rate of direct learning is 0.05 and the base learning rate of multi-task deep learning method while fine-tuning is 0.008. In addition, the dimension of face features is 1,024. The input ratio for each batch among different datasets is the same.

4.3 Results and Analysis

In this section, we will introduce the testing protocols then show and analysis results in our experiments.

The testing set of CCFD contains 91,904 images of 301 subjects and then it was divided into two parts named Target set and Query set. We chose 50% images from every identity in CCFD testing set randomly as Query set and the rest images are as Target set. We evaluate accuracy according to similarity matrix Sim, where $Sim(i, j)$ represents the similarity of the i-th image in Query set and the j-th image in Target set. By calculating verification rate under different false acceptance rates, we can judge the performance of a model. In our experiments, the accuracy of CCFD testing rate is the verification rate when false acceptance rate equals to 0.1%.

Table 1. The accuracy on LFW.

Deep method	Dataset of pre-train	Learning method	Accuracy
Resnet-VIPL	MS	None	**99.23%**
Resnet-VIPL	MS	Fine-tune	98.38%
Resnet-VIPL	MS	Multi-task	**99.13%**

Table 2. The accuracy on CCFD.

Deep method	Dataset of pre-train	Learning method	Accuracy
Resnet-VIPL	MS	None	64.28%
Resnet-VIPL	MS	Fine-tune	69.46%
Resnet-VIPL	MS	Multi-task	**71.88%**

Table 1 shows that the pre-trained model has the best performance than straightforward fine-tuning and multi-task learning. From the results, it is obvious that the learning method with straightforward fine-tuning drops a lot in the accuracy of LFW. At the same time, the accuracy drops a little with the multi-task method. It shows that multi-task deep learning method has a better performance than straightforward fine-tuning. Table 2 shows the accuracy on CCFD of pre-trained model with different learning methods. We can find that multi-task deep learning gets the best performance. Although with fine-tuning method the accuracy on CCFD improves 5.18% compared to pre-trained model, multi-task deep learning method improves 7.6%. The results show the advantages of multi-task deep learning method on both tasks.

Table 3. The accuracy on LFW and CCFD with different learning methods compared to the pre-trained model.

Deep method	Learning method	Accuracy on LFW	Accuracy on CCFD
Resnet-VIPL	None	-	-
Resnet-VIPL	Fine-tune	−0.85%	+5.18%
Resnet-VIPL	Multi-task	**−0.10%**	**+7.60%**

According to the comparison in Table 3, it is obvious that multi-task deep learning method has the best performance on both tasks. First, the accuracy of fine-tuning directly is less than the accuracy of multi-task deep learning method on LFW and CCFD test datasets. Second, although the accuracy on LFW of our approach is less than without fine-tuning, the gap is so small and the accuracy on CCFD of approach is much higher than without fine-tuning. In total, experiments prove that our approach is easy to transfer and it achieves the best results compared to methods without fine-tuning and straightforward fine-tuning. By combining multi-task deep learning and fine-tuning method, our method makes full use of minute large-scale dataset that we can obtain and we also can get a satisfactory model works on both tasks in good performance.

5 Conclusions

In this paper, we proposed a MTDL method for face recognition. Datasets with diverse properties are considered as different tasks. MTDL firstly pre-trains

a deep model on a large-scale face dataset. Then by combining multiple face datasets as the input of the deep model, MTDL shares the same hidden layers while learns task-specific fully-connected layers for different datasets. Experiments on LFW and CCFD datasets show that MTDL generalizes well on different face databases and achieves better performance than the straightforward fine-tuning strategy.

Acknowledgements. This work was supported by the National Program on Key Basic Research Project under Grant 2013CB329304, the National Natural Science Foundation of China under Grants 61502332, 61432011, 61222210.

References

1. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E., et al.: Labeled faces in the wild: a database forstudying face recognition in unconstrained environments. In: Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition (2008)
2. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)
3. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)
4. Caruana, R.: Multitask learning. In: Thrun, S., Pratt, L. (eds.) Learning to Learn, pp. 95–133. Springer, Heidelberg (1998). doi:10.1007/978-1-4615-5529-2_5
5. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 94–108. Springer, Cham (2014). doi:10.1007/978-3-319-10599-4_7
6. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems (NIPS), pp. 1988–1996 (2014)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
8. Zhang, X., Zhou, F., Lin, Y., Zhang, S.: Embedding label structures for fine-grained feature representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1114–1123 (2016)
9. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
10. Long, M., Wang, J.: Learning multiple tasks with deep relationship networks. arXiv preprint arXiv:1506.02117 (2015)
11. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In: European Conference on Computer Vision, Springer (2016)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4179-7
ISSN 1459-2045